

EgdeAI-Learn – Eksplorativ Produktutvikling

JOSTEIN ENGESÆT

HASSAN HEIDARI

FRIDA LILLESALT VALVIK

HANS CHRISTIAN MORKA

IVAR MICHAEL TANGERAAS

VEILEDER

Hallgeir Nilsen

Universitetet i Agder, 2024

Fakultet for Samfunnsvitenskap

Bachelor



Forside

IS-304: 2024

Tittel: EgdeAI-Learn – Eksplorativ Produktutvikling

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen & Geir Inge Hausvik
Veileder	Hallgeir Nilsen
Oppdragsgiver:	Egde Consulting AS

Studenter:

Etternavn	Fornavn
Engesæt	Jostein
Heidari	Hassan
Lillesalt, Valvik	Frida
Morka	Hans Christian
Tangeraas	Ivar Michael

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA X	NEI
Kan besvarelsen brukes til undervisningsformål?	JA X	NEI
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA X	NEI

Forord

Dette dokumentet representerer avslutningen på tre fulle skoleår ved bachelorprogrammet IT & Informasjonssystemer ved Universitet i Agder. Gjennom denne tiden har vi som gruppe fordypet oss i en rekke emner, metoder og teknologier innenfor informatikk og informasjonsteknologi. Denne rapporten utgjør høydepunktet av våre studier, og den representerer vårt engasjement, vår læring og vår innsats for å utforske og bidra til feltet.

I samarbeid med bedriften Egde Consulting har vi hatt privilegiet av å utvikle en AI-applikasjon med mål om å skape verdi for Egdes voksende bibliotek av AI-baserte verktøy og tjenester. Dette prosjektet har vært en unik mulighet til å anvende vår teoretiske kunnskap og kompetanse i en praktisk setting og bidra med løsninger som har reell betydning. Vi har samlet våre kunnskaper, erfaringer og innsikter for å presentere et omfattende bilde av vårt prosjektarbeid, våre funn og våre refleksjoner.

Som en del av dette prosjektet har vi hatt gleden av å samarbeide med dedikerte veiledere, ressurssterke organisasjoner og inspirerende medstudenter og kolleger. Deres bidrag har vært uvurderlig for vår utvikling og for kvaliteten på både denne rapporten og den utviklede applikasjonen.

Vi ønsker å rette en hjertelig takk til vår veileder, førstelektor Hallgeir Nilsen, som har bidratt med gode diskusjoner og samarbeid underveis i prosjektet. Vi vil også rette en takk til oppdragsgiver Øyvind Brekkhus Sandåker og Egde for muligheten til å utføre bachelorarbeidet sammen med dere.

Hjertelig takk til våre familier og nære for deres støtte, oppmuntring og tålmodighet gjennom hele denne reisen. Deres tro på oss har vært motiverende og uvurderlig, og vi er takknemlige for deres støtte gjennom prosessen.

Vi håper at vårt arbeid vil bidra til videre utforskning, diskusjon og innovasjon innen AI og IT-feltet. Denne rapporten er et resultat av vårt felles engasjement for å forstå og forme fremtiden for teknologi og informasjonssystemer.

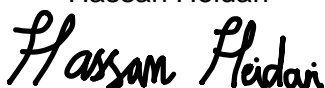
Takk for oss, og vi ser frem til å dele vårt arbeid med dere.

Gruppe 9, 15.05.24, Kristiansand

Jostein Engesæt



Hassan Heidari



Frida Lillesalt Valvik



Hans Christian Morka



Ivar Michael Tangeraaas



Sammendrag

Gjennom bachelorsamarbeidet hos Egde Consulting våren 2024 har bachelorgruppen arbeidet med mål om å bidra til utviklingen av flere AI-baserte verktøy og tjenester under det som omtales som "AI i Egde" eller EgdeAI. Ved utformingen av samarbeidsavtalen med Egde høsten 2023 ble gruppen presentert med en nær frittstående bacheloroppgave, med kun krav om å "gjøre noe gøy med AI". Resultatet av oppgaveutformingen og samtalene mellom interessentene i det kommende samarbeidet førte til fremveksten av visjonen for bachelorprosjektet - EgdeAI-Learn.

I dag har produktet følgende visjon og beskrivelse:

“EgdeAI-Learn har som mål å effektivisere måten ansatte i et selskap samler bedriftsrelaterte data og kunnskap. Enten du er nyansatt eller har vært på arbeidsplassen i flere år, har Learn som mål å trekke ut informasjon for brukeren gjennom en samtalebasert utveksling av informasjon og data. Så enten du er en nyansatt som er nysgjerrig på utviklingsmiljøene teamet ditt jobber med, eller en veteran på arbeidsplassen som kanskje ønsker å oppfriske hukommelsen om et bestemt emne, er Learn her for å hjelpe.”

Bachelorrapporten har som mål å belyse den helhetlige arbeidsprosessen som fant sted våren 2024. Dette inkluderer hvordan bachelorgruppen har arbeidet med en så åpen bacheloroppgave, gruppens smidige arbeidsprosess, valg av verktøy og tjenester, kvalitetssikring, kompetanseutvikling med mer. Rapportens struktur legger til rette for å danne forståelse rundt både arbeidsprosess og produktutvikling gjennom hele bachelorløpet.

Demo presentasjon

[Klikk her](#), eller gå til følgende lenke for å se en demo presentasjon av EgdeAI-Learn i bruk:

<https://www.youtube.com/watch?v=jb03kZ6dmdA>

Innholdsfortegnelse

Forord	3
Sammendrag	4
Demo presentasjon	4
Innholdsfortegnelse	5
Figurliste	8
1. Introduksjon	9
1.1 Sentrale begreper	9
1.2 Bakgrunn og motivasjon	10
1.3 Oppgavebeskrivelse	10
1.4 Visjoner og mål	11
1.4.1 Læringsutbytte og verdiskaping	11
1.4.2 Definisjon av kvalitet.....	11
1.4.3 Behandling av kunnskapsdata	12
1.4.4 Chatbot, prompt profilering og begrensninger.....	13
2. Prosjektstyring	14
2.1 Prosjektplanlegging	14
2.2 Restriksjoner ved prosjektløpet	14
2.3 Arbeidsstruktur og samarbeid	15
2.4 Arbeidsmetodikk – å jobbe agilt	16
2.4.1 Scrum roller.....	17
2.4.2 Agile scrum artefakter	18
2.4.3 Scrum aktiviteter	19
2.5 GitHub Projects	23
2.6 Estimering	25
2.7 Risikoanalyse	26
2.8 Valg av verktøy og teknologi	27
2.8.1 UX-verktøy	27
2.8.2 Frontend verktøy	27
2.8.3 Backend verktøy.....	27
3. Prosjektgjennomføring	29
3.1 Kartlegging av brukerbehov	29
3.1.1 Målgruppen.....	29
3.1.2 Personas.....	30
3.1.3 Empatikart	30
3.1.4 Brukerhistorier	31
3.2 Kartlegging av systemkrav	31
3.2.1 Systemdefinisjon	31
3.2.2 MoSCoW	32

3.2.3 Prioritering av systemkriterier	34
3.3 Beslutninger som formet prosjektet	34
3.3.1 Tidlig designendringer	34
3.3.2 Administratorside	35
3.3.3 Arkivering av chat-historikk	35
3.4 Design av brukergrensesnitt	36
3.4.1 Designinspirasjon	36
3.4.2 Skissering.....	36
3.4.3 Wireframes	37
3.4.4 Prototype / mockup.....	38
3.4.5 Grafiske designprinsipper og retningslinjer	38
3.5 Implementering av systemet (Front- og Backend).....	41
3.5.1 Systemarkitektur	41
3.5.2 Løs kodestruktur	42
3.5.3 Frontend.....	43
3.5.4 Backend	44
3.5.5 Implementeringsutfordringer og sikkerhetstiltak.....	45
3.5.6 Fremtidige utviklingsmuligheter	46
4. Kvalitetssikring	47
4.1 Mål for kvalitetssikring	47
4.1.1 Produktet tilfredsstillende forventninger	47
4.1.2 Fornøyde brukere	48
4.1.3 Kvalitet i kode	48
4.2 Testing/evaluering av design.....	49
4.3 Kvalitetssikringsaktiviteter.....	50
5. Refleksjon	52
5.1 Hva har gått bra? (og hvorfor)	52
5.1.1 Arbeidsmetodikk.....	53
5.1.2 Kontinuerlig dokumentering av arbeid.....	53
5.1.3 Samarbeid med Egde	54
5.2 Hva kunne vi gjort bedre? (utfordringer og begrensninger).....	54
5.2.1 Gjennomgang av kode	54
5.2.2 Tilgjengelighet fra Egde.....	55
5.2.3 Estimering og faktisk tidsbruk.....	56
5.3 Hva har vi lært & hva ville vi gjort annerledes?.....	56
5.3.1 Hva har vi lært?.....	56
5.3.2 Hva ville vi gjort annerledes?.....	57
6. Avsluttende ord.....	59
7. Litteraturliste.....	60
8. Vedlegg.....	62

Vedlegg 1 - Uttalelse fra oppdragsgiver	62
Vedlegg 2 - Egenvurdering	63
Vedlegg 3 - Gruppekontrakt	65
Vedlegg 4 - Prosjektbeskrivelse	67
Vedlegg 5 - Sprint Planning	74
Vedlegg 6 - Daily Scrum	76
Vedlegg 7 - Sprint Retrospective	77
Vedlegg 8 - Ukeslogg 10	84
Vedlegg 9 - Estimeringsgraf #1 - Estimert arbeidstid per Sprint	86
Vedlegg 10 - Estimeringsgraf #2 - Registrert arbeidstid per Sprint	86
Vedlegg 11 - Risikoanalyse	87
Vedlegg 12 - Spørreskjema	88
Vedlegg 13 - Personas	90
Vedlegg 14 - Empatikart	92
Vedlegg 15 - Brukerhistorier	94
Vedlegg 16 - Visuell brukerhistorie	95
Vedlegg 17 - Factor-modell	96
Vedlegg 18 - MoSCoW-analyse (original og oppdatert)	97
Vedlegg 19 - Systemkriterier	98
Vedlegg 20 - Sketches	99
Vedlegg 21 - Lo-Fi Wireframes	100
Vedlegg 22 - Hi-Fi Wireframes	101
Vedlegg 23 - Prototype Chat	104
Vedlegg 24 - Prototype admin page	106
Vedlegg 25 - Definisjon av Benyons designprinsipper	109
Vedlegg 26 - Eksempler på bruk av Benyons designprinsipper	110
Vedlegg 27 - WCAG rapport	115
Vedlegg 28 - WAVE evaluering	126
Vedlegg 29 - Systemarkitektur	127
Vedlegg 30 - Brukertest	128
Vedlegg 31 - A/B brukertest	129
Vedlegg 32 - Referat styringsgruppemøte	131

Figurliste

Figur 1 EgdeAI-Learn - Funksjonalitetsløkke	12
Figur 2 Tretrinns retrospective i Miro	22
Figur 3 Skjermdump fra Github Projects	23
Figur 4 Skjermdump fra Github Projects	24
Figur 5 Burn up graf fra Github Projects.....	24
Figur 6 Risikomatrise	26
Figur 7 Persona Elisabeth Grønnskog.....	30
Figur 8 Persona Mads Mikkelsen	30
Figur 9 Empatikart.....	30
Figur 10 Brukerhistorier	31
Figur 11 Visuell brukerhistorie av nyansatt.....	31
Figur 12 Original MoSCoW analyse.....	32
Figur 13 Oppdatert MoSCoW analyse	33
Figur 14 Prioritering av systemkriterier	34
Figur 15 Skisser av chat og administrator-siden.....	37
Figur 16 Low Fidelity Wireframe av chat i Figma.	37
Figur 17 Prototype av chat og administrator-siden i Figma.	38
Figur 18 Benyons designprinsipp med score.....	39
Figur 19 Oversikt av resultat WCAG rapport	40
Figur 20 Systemarkitektur (mer oversiktlig bilde i vedlegg 29)	42
Figur 21 Forslag til fremtidige utviklingsmuligheter i filoversikt.....	46

1. Introduksjon

Dette kapitlet vil gi en introduksjon til prosjektet, og tar for seg bakgrunn og motivasjon, oppgavebeskrivelse, visjoner, og mål gruppen har knyttet til prosjektet. Rapporten inneholder en rekke kritiske begreper som er essensielle for en grundig forståelse av emnet. Disse nøkkelbegrepene vil bli definert og utforsket grundig i delkapitlet for sentrale begreper, da de danner grunnlaget for å kunne forstå ulike deler av oppgaven på en mer inngående måte.

1.1 Sentrale begreper

EgdeAI – EgdeAI kan defineres som Egde Consultings voksende produktlinje av AI-baserte verktøy og tjenester. Ved tidspunktet for bachelorarbeidet er EgdeAI-Chat hovedproduktet som tilbys innenfor EgdeAI "paraplyen".

EgdeAI-Chat – en avansert chatbot utviklet av Egde, basert på OpenAIs GPT-teknologi. Chatboten er designet for å forstå og svare på forespørsler på flere språk, og tilbyr intelligent og relevant informasjon basert på brukerens spørsmål.

Kunnskapseier – En kunnskapseier er en administrator i systemet. En kunnskapseier har som jobb å legge til og oppdatere nødvendig informasjon i en kunnskapsbase. En kunnskapseier kan ha ansvar for en eller flere kunnskapsbaser. Nøkkelpersoner med tilgang på/oversikt over relevante data vil i stor grad være mest passende i denne rollen.

Kunnskapsbase – En kunnskapsbase er lagringsplassen for relevante bedriftsdata. Ved opprettelse av en kunnskapsbase kan administratorer/kunnskapseiere i systemet laste opp relevante data til en gitt kunnskapsbase. Ordinære brukere av systemet kan da starte chatter med en av disse kunnskapsbasene ved å velge et samtaleemne i chatten. Konteksten for denne samtalen vil være basert på de opplastede dataene.

Mono repo – refererer til en samlet kodebase for flere applikasjoner og pakker. Innenfor EgdeAI finnes både EgdeAI-Chat og EgdeAI-Learn i samme Github-repositorium, kjent som et mono repo.

Embedding – er en prosess hvor data, vanligvis i form av tekst eller bilde, blir oversatt til vektorverdier.

Vektorverdier – er et sett av tallverdier, anvendt for å representere avanserte objekter som ord, bilder, videoer osv. Vektordatasett utgjør en uunnværlig komponent for AI-oppgaver (IBM, n.d.). I denne konteksten vil vektorverdiene være en representant av dataen til kunnskapsbasen som blir brukt for likhetssøk av AI-en for å finne relevant informasjon til å svare på brukerens prompt.

Blob – (Binary Large Object) representerer en omfattende mengde av uforanderlig rådata, som blir lest i binær form eller som tekst (MDN Web Docs, 2023). Innenfor den gitte konteksten blir blob dataen lagret i en skybasert lagringsinfrastruktur – Azure Blob Storage.

LLM – (large language model) er dype læringsalgoritmer som demonstrerer evnen til å identifisere, oppsummere, oversette, prognostisere og generere innhold ved å utnytte omfattende datasett (NVIDIA, n.d.).

Prompt profilering – (prompt profiling) er et regelsett som setter retningslinjer for konteksten AI-en skal svare ut ifra. Brukes ofte for å sette opp personligheten og rollen til AI-en, altså at den er klar over hvem den skal være og oppføre seg som. Andre kritiske kontekst data kan også defineres her.

1.2 Bakgrunn og motivasjon

Før introduksjonen av emnet IS-304, samarbeidet gruppen i emnet "Prosjektgjennomføring - IS-310". IS-310 fungerte som en forberedelse til bacheloremnet IS-304. Gjennom dette emnet utforsket gruppen potensielle prosjektmuligheter hos bedrifter med sikte på den påfølgende bacheloroppgaven i IS-304. Det var en felles oppfatning blant gruppen at IS-310 og dets arrangementer, som bedriftsmøte-seminaret RefreshIT, ville spille en betydelig rolle i utformingen av bacheloroppgaven og det kommende vårsemesteret. Før et eventuelt valg av bachelorprosjekt eller samarbeid, ble gruppen enige om at bacheloroppgaven måtte gi muligheter for kompetansevekst for alle medlemmene i bachelorgruppen. Dette innebar at den valgte bacheloroppgaven måtte tilby muligheter for vekst innenfor frontend, backend og UX-arbeid.

Etter grundige samtaler med ulike bedrifter under og etter RefreshIT, besluttet gruppen å inngå et samarbeid med Egde Consulting AS (heretter referert til som Egde).

1.3 Oppgavebeskrivelse

Egde presenterte sitt forslag til bacheloroppgave under RefreshIT høsten 2023. Forslaget som ble lagt frem var en svært åpen arbeidsoppgave med ønske om videre forskning og utvikling av EgdeAI. Egde uttrykte et ønske om å utvide sin forståelse og bruk av AI ved å integrere flere AI-drevne verktøy under en paraply som kunne gagne bedriften som helhet. Konseptet og visjonen for "AI i Egde" ble overlatt til gruppens utforskning. Gruppen hadde på forhånd uttrykt en viss skepsis mot bacheloroppgaver som kunne virke "for åpne", men begynte raskt å se potensialet for en bacheloroppgave som kunne tilby de ønskede arbeidsmulighetene som gruppen var på utkikk etter. I kontakt med Egde ble gruppen utfordret til å formulere sin egen oppgavebeskrivelse som passet inn med Egdes ønske og visjon for videre forskning, samtidig som den skulle samsvare med UiAs kriterier for en gitt bacheloroppgave.

Etter en effektiv idemyldringsprosess og diskusjon internt i gruppen kort tid etter møtet med Egde, ble det lagt frem et konsept for det som skulle bli EgdeAI-Learn. En AI-basert tjeneste/verktøy som skulle bidra til å effektivisere nyansettelsesprosesser, onboarding og generell informasjonsinnhenting innad i bedriften (se vedlegg 4 for den fulle oppgavebeskrivelsen). Egde og gruppen ble samme dag enige om å inngå et bachelorsamarbeid for å realisere konseptet for EgdeAI-Learn.

1.4 Visjoner og mål

Visjonen til prosjektet og ideen bak EgdeAI-Learn har vært under kontinuerlig justeringer gjennom prosjekts varighet. I starten av prosjektet ble visjonen for EgdeAI-Learn sitt utfall formet uten en fullstendig forståelse av hvilke mål som var realistiske å oppnå innenfor en tidsramme på fem måneder. Underveis måtte gruppen ta valg som å endre systemet, både legge til nye funksjoner, og fjerne andre. Den første visjonen av EgdeAI-Learn var at verktøyet skulle kun bli utviklet for nyansatte i en bedrift, altså et verktøy som skulle få ansettelsesprosessen i en bedrift til å bli mer effektivisert. Ved senere anledning ble det vedtatt at EgdeAI-Learn kunne bli tilbudt til alle ansatte innenfor en bedrift. Utenom dette var det en del andre visjoner og mål som fulgte med gjennom prosjektløpet.

1.4.1 Læringsutbytte og verdiskaping

Som nevnt tidligere, utviklet gruppen en prosjektbeskrivelse høsten 2023 (vedlegg 4) i forkant av bachelorarbeidet som skulle starte våren 2024. I denne prosjektbeskrivelsen definerte gruppen to viktige punkter for egen læring og motivasjon for prosjektarbeidet:

- **At arbeidet resulterer i et produkt/rammeverk som gir verdi til oppdragsgiver/prosjektkunde** - Det var helt sentralt at arbeidet og produktet skulle kunne gi en konkret og reell verdi for samarbeidspartneren. I dette scenarioet ville det bety at systemet skulle kunne bidra til effektivisering av læring for nye og eksisterende ansatte ved en bedrift som tar i bruk produktet. Ikke minst skulle resultatet være en del av videreføringen og utviklingen av AI som fagfelt hos Egde.
- **Gruppe 9s egne læringsutbytte ved prosjektarbeidet** – For gruppe 9 var det helt sentralt at det valgte prosjektet skulle gi verdi i ny læring og arbeidserfaring i en reell situasjon. Prosjektet hos Egde ville by på mye ny læring av ukjente systemer, komponenter, prosesser og teknologier som kunne styrke gruppens og hvert individs kompetanse innen fagfeltet, samtidig som man ville videreutvikle allerede etablert erfaring og kompetanse.

(Engesæt et. al., 2023)

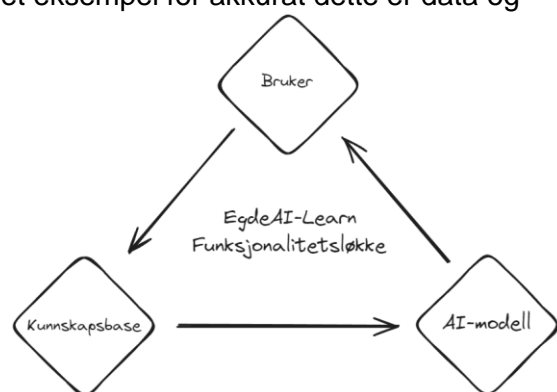
1.4.2 Definisjon av kvalitet

Når det kommer til visjoner og mål relatert til kvalitet, er det nødvendig å beskrive hvordan gruppen definerte kvalitet. Denne definisjonen ble sett på som mål for kvalitetssikringen av prosjektet. Gundersen & Halbo (2018) definerer kvalitet for en gjenstand eller tjeneste slik: «evnen til å tilfredsstillte brukerens krav og forventninger». I forhold til dette prosjektet var det flere aspekter som ville inngå i hvordan kvalitet ble definert. Ettersom produktet ble utviklet i samarbeid med Egde, ble også kravene og forventningene fra oppdragsgiver vektlagt i denne definisjonen. På bakgrunn av dette valgte gruppen å fokusere på tre hovedaspekter for å definere kvalitet:

- **Produktet tilfredsstillende forventninger** - For å sikre kvalitet var det avgjørende at applikasjonen skulle kunne utføre de forventede oppgavene på en tilfredsstillende måte. I dette prosjektet var de spesifikke forventede funksjonene definert internt av prosjektgruppen, da oppdragsgiver satt få presise krav til det endelige produktet. Gruppen fikk mulighet til å utforske og definere oppgaven selv. De konkrete forventningene til applikasjonen ble dermed fastsatt av gruppen og videre diskutert med oppdragsgiver. For å få oversikt over de ulike funksjonene som var ønsket å implementere, kunne gruppen ta i bruk systemdefinisjon og MoSCoW-metoden for deretter å diskutere disse med Egdes forventninger. Disse verktøyene fungerer som hjelpemiddel for å vurdere oppnådd kvalitet.
- **Fornøyde brukere** - En annen viktig faktor for å sikre kvalitet er brukertilfredshet hos målgruppen for applikasjonen. Det er ikke tilstrekkelig at produktet oppfyller tekniske krav og forventninger alene; det må også være intuitivt, brukervennlig og gi en positiv opplevelse for sluttbrukeren. Dette kan måles på flere nivåer, for eksempel ved å undersøke i hvilken grad brukeren finner responsen fra AI-en nyttig.
- **Kvalitet i kode** - For å sikre kvalitet i koden skulle det legges vekt på struktur, effektivitet, lesbarhet, vedlikeholdbarhet, og muligheten til å implementere nye funksjoner. I prosjektbeskrivelsen (vedlegg 4) ble det beskrevet muligheten til å tilby applikasjonen som et produkt til kunder. For å fasilitere for en god overleveringsprosess ville kvalitet i koden vurderes som en sentral rolle. Dersom applikasjonen skal selges videre var det ønsket å ha muligheten til å gjøre endringer i koden for å tilrettelegge for de spesifikke kundene.

1.4.3 Behandling av kunnskapsdata

Gruppens visjon og målsetting var å utvikle en pålitelig og effektiv AI-løsning som innhenter og formidler kunnskap om emner valgt av bedriften selv. Dette muliggjør ansatte å stille spørsmål til AI-en basert på den kunnskapsbasen den besitter. Et konkret eksempel for akkurat dette er data og kunnskap relatert til en personalhåndbok. For eksempel ville en kunnskapseier av bedriftens personalhåndbok være en person tilknyttet HR-avdelingen. Ved å la HR-avdelingen laste opp ønsket informasjon om personalhåndboken til en dedikert kunnskapsdatabase med navnet "Personalhåndboken", kan AI-en holde informasjonen tilgjengelig for ansatte. Dette gir ansatte valget mellom å lese gjennom omfattende dokumentasjon for å finne svar på spørsmål om for eksempel feriedager eller andre ansattrettigheter, eller å stille spørsmål direkte til AI-en. En ansatt kunne da ha spurt "Hvor mange feriedager har jeg krav på?", hvor AI-en vil gi svar basert på tilgjengelig informasjon i kunnskapsbasen.



Figur 1 EgdeAI-Learn - Funksjonalitetsløkke

1.4.4 Chatbot, prompt profilering og begrensninger

Det er viktig å presisere at å jobbe med AI ikke nødvendigvis betyr at det er planlagt å utvikle en egen AI-modell fra grunnen av. Prosessen med å utvikle og trene en egen AI-modell ville vært et helt separat prosjekt som krever betydelige ressurser, inkludert datakraft, treningsdata og ekspertise, som gruppen ikke besitter. Heldigvis eksisterer det en rekke utviklede AI-modeller som kunne brukes til prosjektets formål. Gjennom dialog og erfaring fra Egde ble det bestemt at hjernen bak chat-tjenesten til EgdeAI-Learn skulle drives av OpenAIs GPT-modeller.

Rett ut av boksen vil en GPT-modell være ferdig opplært med omfattende kunnskap, gode samtale- og formuleringsevner, og generelt alt man er vant med fra bruk av verktøy som ChatGPT. For den planlagte hensikten til EgdeAI-Learn var det samtale- og formuleringsevnen som var sentralt for å oppnå ønsket resultat. Dataene og kunnskapen som chatbot-tjenesten skulle svare ut ifra ville være basert på dataene fra kunnskapsbasene, og i mye mindre grad komme fra den opptrente kunnskapen til GPT-modellen. Det var også sentralt at chatbot-tjenesten skulle ha en personlighet som samsvarte med visjonen bak EgdeAI-Learn. For å oppnå dette, ville prompt profilering tas i bruk, der AI-en sin oppførsel ville bli definert. Her kunne også litt basis informasjon oppgis, slik at AI-en kunne ha noe standardisert innhold å svare ut ifra, uansett hvilken kunnskapsbase som ble snakket med. Dette kunne være generell informasjon om AI-ens oppgaver og gjøremål, informasjon om Egde og lignende.

Dataen som ble lastet opp i kunnskapsbasene ville definere hva AI-en kunne svare på. Denne begrensningen er en av de største motsetningene mellom EgdeAI-Learn og andre eksisterende generative LLM tjenester som ChatGPT og EgdeAI-Chat, da disse kan svare på alle mulige spørsmål, fordi GPT-modellene har et enormt datagrunnlag å svare ut ifra.

Det er viktig å igjen poengtere at EgdeAI-Learn fortsatt skulle presentere svaret i et samtalebasert format, ikke bare hente ut dataen. Kort oppsummert skulle EgdeAI-Learn være en generativ LLM tjeneste med en begrensning på kunnskapen de genererte svarene kunne basere seg på, altså dataene fra kunnskapsbasene. For å oppnå en slik begrensning, og en egendefinert EgdeAI-Learn profil, ville prosjektet være avhengig av denne prompt profileringen. Derfor ville en tydelig og godt strukturert prompt profilering spille en vesentlig rolle i AI-ens oppførsel, begrensninger og responsmønster.

2. Prosjektstyring

Dette kapitlet av rapporten har som mål å presentere betydningen av prosjektstyring, metodene, og verktøyene som er blitt brukt gjennom bachelorarbeidet. Godt integrert prosjektstyring er essensielt for å sikre suksess fra start til slutt i prosjektet. For å oppnå ønskede resultater er det avgjørende å gjennomføre grundig analyse, refleksjon av arbeid, og identifisere de viktigste faktorene som bidrar til prosjektets suksess.

2.1 Prosjektplanlegging

Ved oppstartsfasen av ethvert prosjektforslag er grundig planlegging avgjørende for å sikre en vellykket gjennomføring. Prosjektplanleggingen representerer den første og kanskje viktigste fasen av et prosjekt. Det er her grunnlaget for hele prosjektet legges, og hvor nødvendige strategier og tiltak blir utviklet for å oppnå definerte mål.

En grundig planlagt prosjektplan gir ikke bare et detaljert veikart for å nå målene, men den fungerer også som et avgjørende verktøy for å koordinere og styre alle aktiviteter og ressurser som er involvert. Uten en solid plan risikerer prosjektet å miste retning, oppleve forsinkelser, overskridelser av ressurser og til og med mislykkes fullstendig.

I kontekst av dette bachelorprosjektet og oppgavens åpne natur, innså gruppen tidlig at en slik rigid og fullstendig tilnærming til oppstarten av prosjektgjennomføringen ville gjøre det vanskelig å forme og utvikle EgdeAI-Learns visjon på lang sikt. Selv om visjonen for produktet var definert på forhånd, ble gruppen og oppdragsgiver tidlig enige om at forskningen og uttestingen av verktøyene kunne i stor grad endre den endelige målsetningen for sluttproduktet. Derfor ble det midlertidig utarbeidet en intern prosjektstrategi som tilrettela for utforskning og smidig utvikling gjennom hele prosjektløpet. Som en del av denne prosjektstrategien ble det utarbeidet en rekke grafer, diagrammer og risikomatriser for å sikre en felles forståelse av det kommende arbeidet.

For å legge til rette for en smidig prosjektgjennomføring, valgte gruppen å hente inspirasjon fra den agile arbeidsmetodikken scrum. Den valgte arbeidsmetodikken skulle sikre regelmessig oppfølging av kravspesifikasjoner, kvalitet og målsetninger gjennom hele prosjektarbeidet, samtidig som den skulle gi mulighet for fleksibilitet og endringer underveis i prosjektet (se delkapittel 2.4 for mer informasjon om dette).

2.2 Restriksjoner ved prosjektløpet

Selv med en åpen oppgave og en agil tilnærming til prosjektgjennomføringen, var det viktig å ha forståelse for tre grunnleggende elementer som nesten alltid er til stede i ethvert prosjekt: kostnad, tid og omfang. For å effektivt beskrive forholdet mellom disse tre elementene, kan man ta utgangspunkt i *prosjekttrekanten*, også kjent som *triple constraint*. Prosjekttrekanten har som mål å representere forholdet mellom de nevnte elementene i et prosjekt, hvor endringer i ett av disse elementene vil påvirke de to andre (Rolstadås, 2020).

Det er verdt å merke seg at prosjekttrekanten ikke nødvendigvis spilte en avgjørende rolle ved utformingen og planleggingen av dette bachelorprosjektet. Den fungerte heller som et nyttig verktøy for å illustrere sammenhengen mellom de viktige elementene som spilte en betydelig rolle i dialogen og planleggingsfasen av bachelorprosjektet. For det er nettopp sammenhengen mellom disse elementene som legger grunnlaget for å oppnå ønsket resultat og kvalitet i et prosjekt (Rolstadås, 2020).

Elementet knyttet til tid spiller en avgjørende rolle i dette bachelorprosjektet på grunn av den fastsatte tidsfristen for innlevering av prosjektet. Da denne fristen er uforanderlig, må omfanget og kostnaden tilpasses for å oppnå ønsket kvalitet innenfor gitte rammer.

I dette tilfellet har kostnadselementet en annen betydning enn i vanlige prosjekter, da prosjektet gjennomføres som en del av en bachelorutdanning og kostnader primært består av tilgjengelige arbeidstimer for studentene, ikke økonomiske utgifter.

Arbeidsomfanget vil bli påvirket av tilgjengelig tid; gruppen estimerte at de kunne dedikere syv arbeidstimer, fire dager i uken, til prosjektet gjennom hele semesteret. Ekskludert eventuelle sykedager og feriedager vil gruppen ha omtrent 532 arbeidstimer tilgjengelige per student mellom oppstarten ved Egde i januar og tidsfristen for bachelorrapporten i mai.

Dermed hadde gruppen identifisert begrensninger ved to av de tre elementene i prosjekttrekanten. Det var derfor avgjørende å avgrense arbeidsomfanget i tråd med disse funnene for å sikre at prosjektet ville bli levert innen fristen i mai.

2.3 Arbeidsstruktur og samarbeid

Som et prosjektteam har gruppe 9 utviklet erfaring innen prosjektgjennomføring og samarbeid gjennom hele bachelorløpet. Gruppemedlemmene har arbeidet tett sammen over flere semestre for å oppnå gode resultater og læring. Gjennom studiene har gruppen opparbeidet seg solid faglig kompetanse både individuelt og som en samlet enhet.

Gruppen har utviklet en dynamisk arbeidsmetodikk som optimaliserer hver enkeltes evner, erfaringer og styrker, samtidig som den kompenserer for eventuelle svakheter og forbedringsområder. Dette skaper et inkluderende og effektivt arbeidsmiljø som fremmer åpenhet, tillit og gjensidig respekt. Den kollektive sammensetningen av gruppen legger til rette for en atmosfære der alle kan bidra med sine unike tanker og ferdigheter, og deres arbeid utfyller og beriker hverandre.

Arbeidsstrukturen internt i gruppen kan sees på denne måten:

- Prosjektleder og scrum master – Fullstack-utvikling
 - o Ivar Michael Tangeraas
- Frontend og UX-design - team
 - o Frida Lillesalt Valvik
 - o Jostein Engesæt
- Backend - team
 - o Hassan Heidari
 - o Hans Christian Morka

Selv med definerte roller og fokusområder har gruppen samarbeidet og bidratt kollektivt til utformingen av produktet EgdeAI-Learn på tvers av fagområdene. Det er nettopp på grunn av denne dynamikken og fordelingen av ansvar at gruppen, på tvers av kompetanse og erfaring, kunne bidra til å identifisere prosjektets potensialer, utfordringer og flaskehalsen gjennom hele prosjektløpet. Å arbeide med et bachelorprosjekt som var til fordel for hver enkelt medlems interesse og fagområde var som nevnt tidligere topp prioritet før starten av bacheloroppgaven. Dette har vært en hovedgrunn til at samarbeidet og strukturen har fungert optimalt. Å skape den kulturen og troverdigheten som gruppen nyter godt av hver dag, kommer ikke av seg selv, men er et resultat av læring, feiling og aktiv deltakelse fra alle parter.

2.4 Arbeidsmetodikk – å jobbe agilt

Som nevnt i delkapittel 2.1 prosjektplanlegging, valgte gruppen å dra inspirasjon fra den smidige/agile arbeidsmetodikken scrum. I følge *Store norske leksikon* og *Torgeir Dingsøy* er scrum “den internasjonalt mest brukte smidige utviklingsmetoden. Metoden egner seg for små team på opptil ti personer som jobber på heltid. Den ble utviklet for å lage et programvareprodukt i tett samarbeid med kunden.” (2024). Scrum er en arbeidsmetodikk som gruppen har praktisert i sitt arbeid gjennom flere prosjektoppgaver i studiene ved bachelorprogrammet, og har opplevd betydelig nytteverdi av. Denne nytteverdien finnes i metodikkens egenskap til å sikre regelmessig tilbakemelding og justering av prosjektet basert på endrede krav eller nye innsikter. Dette kunne bidra til å minimere risikoen for feilretting sent i prosessen og samtidig økte muligheten for å tilpasse seg eventuelle endringer i prosjektmiljøet, uansett hvor i produksjonsløpet man befant seg.

Det bør derimot påpekes at scrum-metodikken ikke ble fulgt religiøst til punkt og prikke gjennom prosjektløpet. Den bidro heller til utformingen av gruppens egne arbeidsmetodikk. Dette valget ble begrunnet av arbeidsoppgavens åpne natur og en viss grad av uforutsigbarhet som ikke harmoniserte med noen av scrums få waterfall-lignende ritualer og artefakter. Mer om dette diskuteres senere i dette delkapittelet.

Scrum-manifest dekker fire hovedpunkter som skiller den smidige arbeidsmetodikken fra typiske *fossefallsmetoder* innen prosjektgjennomføring (Dingsøy, 2024):

1. «Personer og samspill fremfor prosesser og verktøy.»
2. «Programvare som virker fremfor omfattende dokumentasjon.»
3. «Samarbeid med kunden fremfor kontraktsforhandlinger.»
4. «Å reagere på endringer fremfor å følge en plan.»

Disse fire hovedpunktene har fungert som en katalysator for bachelorarbeidet fra start til slutt. Som beskrevet tidligere, har gruppen erfart styrken av å benytte seg av et godt samspill og dra nytte av individets erfaringer og egenskaper i prosjektarbeidet. På samme måte har gruppen erfart at å faktisk produsere gode systemer har mer verdi enn å bare *drømme* om dem i dokumentasjonen og planleggingen. Det tette samarbeidet med Egde har ført til god idemyldring, tilbakemelding og oppfordring som har hatt sterk nytteverdi for produktet i en helhet. Og sist, men ikke minst, har muligheten til å jobbe smidig og reagere på situasjoner underveis i produksjonen bidratt til at gruppen mer effektivt kunne optimalisere og tilpasse arbeidet etter behov og ønsker.

Gruppen mener selv at det er nettopp disse punktene som har gjort det mulig å produsere, lære og erfare en vellykket gjennomføring av prosjektet.

2.4.1 Scrum roller

Innen scrum-metodikken innebærer det et sett med forskjellige roller og ansvarsområder som bidrar til å skape et oversiktlig og kontrollert miljø for utviklingsarbeidet. Scrum.org anbefaler at et scrum-team bør bestå av en liten gruppe mennesker, maksimalt 10 personer, som sammen bidrar til å nå produktmålet gjennom en ikke-hierarkisk, profesjonell enhet (u.å.). De tre mest sentrale rollene, eller ansvarsområdene, i en scrum-prosess er:

1. Produkteier
2. Scrum Master
3. Utviklerteamet

Ansvarsområdet til en produkteier omhandler hovedsakelig å ivareta produktet eller prosjektets visjon gjennom utforming og oppfølging av backloggen (Scrum.org, u.å.). Scrum-masteren har ansvar for å sikre at scrum-teamet kan jobbe smidig gjennom hver etappe eller sprint (Scrum.org, u.å.). Utviklerteamet kan defineres som de resterende medlemmene av scrum-teamet som utgjør store deler av arbeidskraften ved gjennomføring av planlagte backlogg-artikler og arbeid i prosjektet.

I konteksten av dette bachelorprosjektet smelter noen av disse rollene og ansvarsområdene mer sammen. For eksempel ble rollene og ansvarsområdene for både produkteier og scrum master sammenslått og kun omtalt som scrum master. Det er viktig å påpeke at arbeidet med utforming og oppfølging av backloggen ikke har blitt utført av scrum masteren alene, men heller i stor grad som en samlet enhet.

Hele scrum-teamet har bidratt til opprettelse, utforming og oppdatering av backlogg-artikler, men scrum master har hatt ekstra ansvar for oppfølging og tilbakemelding på strukturen og progresjonen. Scrum-arbeidet fungerer, som sagt, best når alle bidrar til arbeidet mot et felles produktmål.

I dette tilfellet kunne produkteieren vært en person fra Egde, for eksempel gruppens interne veileder, Øyvind Brekkhus Sandåker, men som en del av læringsutbyttet og oppmuntring fra veilederen har Egde hatt en mer passiv rolle når det kommer til prosjektgjennomføringen og scrum-arbeidet, og har heller fungert som interessent (stakeholder). Dialogen med veilederen og Egde har hovedsakelig fungert via ukentlige show-and-tell-møter og veiledning etter behov. Dette antyder at scrum-arbeidet i stor grad har blitt gjennomført og fulgt opp internt i gruppen, før fremdrift og prosess rapporteres til nødvendige eksterne parter som Egde og UiA.

2.4.2 Agile scrum artefakter

Det finnes tre sentrale scrum-elementer som bidrar til å måle fremgang, kvalitet og retning innenfor scrum-rammeverket:

1. Produktbacklogg
2. Sprintbacklogg
3. Produktinkrement

Disse elementene danner grunnlaget for en slags blåkopi av produktets helhetlige struktur og visjon, som effektivt kan segmenteres og behandles i mindre, konsentrerte biter. Disse artefaktene, sammen med senere nevnte scrum-aktiviteter, utgjør noe av grunnlaget for hvorfor gruppen mener scrum fungerer så effektivt. Samtidig er det viktig å understreke at det er nettopp her gruppen valgte å gå sin egen vei ved dannelsen av sin egne arbeidsmetodikk. Til tross for den etablerte strukturen i scrum-rammeverket, valgte gruppen å forme sin egne agile arbeidsstruktur mer tilpasset oppgavens åpne natur og dets behov og prosjektkrav. Det beste eksemplet på akkurat dette er hvordan Sprint backlogg har spilt en stor rolle ved utformingen av produkt backloggen.

I mange scrum-baserte prosjekter kan *produktbackloggen* effektivt defineres som den faktiske helhetlige blåkopien og visjonen for sluttproduktet som kunden har bestilt. Den blir i stor grad definert i starten av prosjektarbeidet og kartlegger all funksjonalitet som er nødvendig for å utvikle et endelig produkt. Den segmenteres i mange mindre, håndterbare oppgaver og gjøremål, som ofte kalles backlogg-artikler. Selv om mye av dette arbeidet skjer i starten av prosjektarbeidet, er den fortsatt dynamisk og kan raskt endres basert på bestilling og kravendringer (Scrum.org, u.å.). Når første utkast av produktbackloggen er på plass, vil man prioritere hvilke backlogg-artikler som skal utføres først og ta dem med videre inn i scrum-etappearrangementet, sprint, hvor backlogg-artiklene blir en del av det som kalles en *sprintbacklogg*. Denne prioriteringen og utvelgelsen er en kontinuerlig prosess som fortsetter helt til alle backlogg-artiklene er gjennomført og implementert, og den siste sprintetappen er over.

I konteksten av dette bachelorprosjektet har denne prosessen blitt snudd på hodet. Da verken oppgavens oppdragsgiver eller gruppen hadde en klar definert oppfatning av hva sluttproduktet ville være eller se ut som i starten av arbeidet, følte det unaturlig og påtvunget å skulle definere en produktbacklogg for et ukjent produkt. Da forskningen og uttestingen av verktøy og tjenester raskt kunne endre tolkningen og visjonen til produktet, mente gruppen selv at visjonen for sluttproduktet ville ta mer og mer form etter hvert som man beveget seg utover i sprintetappene. Gruppen besluttet derfor heller å danne interne milepæler for produktutviklingen basert på forarbeidet som ble utført og definert ved prosjektplanleggingen. Et godt eksempel på dette er for eksempel spørsmålet om hva som er minimumskriteriet for at produktet skal fungere – en AI-chat-funksjonalitet. Dette bevisstgjorde gruppen om hva som måtte forskes på og utarbeides først, og dannet grunnlaget for de første sprint backlogg-artiklene som ble opprettet. Dermed dannet dette naturlig de første byggsteinene i produktbackloggen.

Denne prosessen fortsatte gjennom hele arbeidsløpet til bacheloroppgaven. Målinger og prioriteringer ble utført ved starten av hver sprintetappe, milepæler for utvikling, kvalitetssikring og funksjonalitetsimplementasjoner ble bestemt, og backlogg-artikler ble opprettet påfølgende. Dette ga gruppen/scrum-teamet større frihet til å utforske muligheter og forme visjonen og sluttproduktet smidig underveis i hele prosjektløpet. Det vil si at en overordnet blåkopi eller produktbacklogg ble dynamisk bygd opp gjennom hver sprint grunnet oppretting av sprintbackloggen. Resultatet av denne tilnærming ved slutten av prosjektet er en fullstendig produktbacklogg med godt dokumenterte backlogg-artikler og refleksjoner som kan brukes til videre refleksjon og ettertanke ved videre arbeid.

Dette ble videre støttet av Egdes oppfordring om å ha frie tøyler til å eksperimentere og bidra til forskningen rundt "AI i Egde". Egde oppfordret også til å fokusere på å implementere nye funksjoner og øke funksjonaliteten kontinuerlig i produksjon, i stedet for å vente til slutten av hver sprint. Dette bidro enda mer til det smidige utviklingsmiljøet gruppen ønsket å jobbe under og gjorde det mulig å kontinuerlig vise til endringer og progresjon ved de faste, ukentlige møtearenaene. Denne kontinuerlige veksten og utviklingen av produktet kan defineres som produktinkrement innenfor scrum-metodikken.

2.4.3 Scrum aktiviteter

Da prosjektarbeidet har hatt en relativt smidig og fri tilnærming til utviklingen av sluttproduktet, har det vært viktig å benytte en rekke scrum-aktiviteter og arrangementer for å sørge for at prosjektet holder stø kurs. Dette har vært for å effektivt måle prosjektets progresjon, reflektere over utført arbeid og sikre at scrum-teamet har en felles forståelse og oppfatning av kommende gjøremål og arbeid.

De mest sentrale aktivitetene som ble brukt under scrum-arbeidet var:

1. Sprintetapper
2. Sprint planning
3. Daily scrum
4. Sprint retrospective
5. Weekly standup

Sprintetappe

Et ord som går igjen når man diskuterer arbeid gjennom scrum er begrepet *sprint*. Gjennom beskrivelsen av gruppens agile arbeidsmetodikk bør det ha kommet frem at sprintetapper er en av de mest sentrale aktivitetene i hele prosessen. Scrum.org definerer sprinter som selve "*hjerteslaget til scrum*" (oversatt fra engelsk, u.å.). En sprintetappe omfatter alle de andre scrum-aktivitetene og har vanligvis en varighet på fire uker eller mindre. Målet med sprintene er å segmentere en lengre arbeidsprosess inn i flere mindre biter og skape rom for mer hyppige tilbakemeldinger og oppfølging på arbeidet som blir gjort.

Basert på tidligere erfaring har gruppen valgt å føre sprintetapper med en varighet på tre uker. Dette begrunnes med at varigheten gir rikelig med rom for utvikling og arbeid med sprintbackloggen uten å nødvendigvis bli forstyrret av hyppigere, repeterende intervaller av sprint-aktiviteter som kan ta mye tid. Videre kan det nevnes at sprintperioden kan tilpasses etter behov. Det kan for eksempel være hensiktsmessig å forkorte sprintintervallene hvis prosjektet står i fare for å feile eller generelt ikke er på spor. Dette vil kreve hyppigere gjennomføring av de underliggende sprint-aktivitetene som fremmer refleksjon, felles forståelse og utforming av arbeidet som har blitt og/eller skal utføres. Gruppen hadde dette i bakhodet, men opplevde ikke nødvendigheten av å ta det i bruk i bacheloroppgaven.

Sprint planning

Starten på hver sprint begynner med å planlegge det kommende arbeidet for de neste ukene gjennom et *sprint planning*-møte. Målet med dette møtet er å kartlegge gjøremålene som skal utføres i løpet av sprinten og å skape en felles forståelse blant hele scrum-teamet (Scrum.org, u.å.). Det er i dette møtet at den tidligere nevnte sprintbackloggen blir utformet.

I konteksten av bachelorprosjektet har sprint planning-møtet som regel funnet sted om morgenen på mandager etter en avsluttet sprint uken før. Møtet begynner med en rask gjennomgang av eventuelle refleksjonspunkter fra forrige sprint som gruppen ønsker å ta med videre eller forbedre i den kommende sprinten. Videre diskuterer gruppen hvilke fokusområder og mål sprinten skal ha, før de går videre til sprint backloggen. Disse vurderingen blir som nevnt tidligere tatt basert på hvor i prosjektløpet man befinner seg og hva som trengs å implementeres i tiden fremover.

Først vurderes backlogg-artiklene som ble videreført fra forrige sprint, og det gis en ny vurdering av prioriteringsgrad og tidsestimat for arbeidsoppgavene. Når de tidligere backlogg-artiklene er behandlet, begynner scrum-teamet å generere nye backlogg-artikler individuelt i 15 minutter basert på det valgte hovedmålet for sprinten, etterfulgt av en felles gjennomgang av hva som ble generert og foreslått. Backlogg-artiklene som genereres her kan omhandle alt fra UX, frontend, backend og UiA-relaterte oppgaver som skal gjennomføres i den gitte sprinten.

Avslutningsvis gjennomføres en runde med planning poker for å etablere en felles tilnærming til den estimerte arbeidstiden for en gitt oppgave (mer om dette under delkapittel 2.6). Dette opplevde gruppen som en god arena for å diskutere oppgavene grundigere og skape en bedre forståelse av backlogg-artikkelen. Et møtereferat ville deretter bli skrevet av scrum-masteren og lagret felles og tilgjengelig for hele scrum-teamet (se vedlegg 5 for eksempel på møtereferat).

Daily scrum

For å opprettholde god oversikt og sikre en stø kurs i prosjektarbeidet gjennomførte gruppen et daglig, 15-minutters møte hver morgen. Daily scrum, eller daily stand-up som gruppen omtalte det, var en møtearena hver morgen hvor scrum-teamet kunne formidle dagens individuelle gjøremål, dele erfaringer og be om hjelp. Hensikten med dette møtet var å sikre god flyt i prosjektarbeidet og raskt identifisere uregelmessigheter eller problemstillinger som kunne oppstå i løpet av en sprintetappe. Scrum-masteren ville lede møtet og føre referat over hva som ble nevnt. Tre faste spørsmål ville bli stilt til hvert medlem i scrum-teamet:

1. Hva oppnådde du i går?
2. Hva planlegger du å gjøre i dag?
3. Er det noen hindringer eller utfordringer i veien?

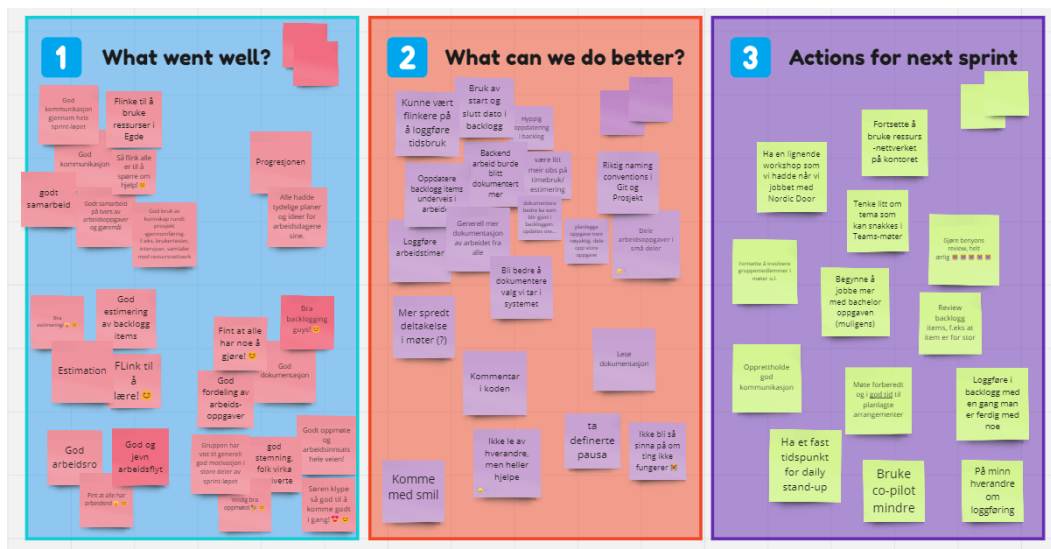
Dette var en god arena for å fordele arbeidskraft på arbeidsoppgavene. Hvis en person opplevde at ekstra fokus eller bistand var nødvendig ved en gitt arbeidsoppgave, kunne dette raskt avdekkes i møtet, og ressurser kunne omfordes der det var hensiktsmessig. Et møtereferat ville deretter bli skrevet av scrum-masteren og lagret felles og tilgjengelig for hele scrum-teamet (se vedlegg 6 for eksempel på møtereferat).

Sprint Retrospective

Ved slutten av hver sprint ble scrum-teamet samlet til et *sprint retrospective*-møte, en refleksjonsbasert samling for å evaluere eget arbeid og diskutere sprintløpet. Møtets mål er å identifisere hva som gikk bra i sprinten og hva som kan forbedres til neste sprint. Kjernen av dette møtet handler om å fremme intern diskusjon og refleksjon blant teammedlemmene.

Gjennom bachelorløpet har hver sprint blitt avsluttet tre uker etter oppstart, på siste dagen i gjeldende uke. Møtet starter med en gjennomgang av backloggen og status på arbeidet som har blitt utført i sprinten. Hvert teammedlem forklarer kort hva som har blitt utført i hver av backlogg-artiklene, etterfulgt av en felles gjennomgang og vurdering av de gjenværende artiklene som ikke ble fullført. Disse artiklene vurderes for om de skal tas med til neste sprint eller nedprioriteres.

Når sprint-backloggen er ferdigbehandlet, gjennomføres en runde med mer refleksjonsbasert diskusjon om gjennomføringen av sprinten. Her har gruppen tatt i bruk en *tretrinns retrospective* analyse for å vurdere *hva som gikk bra, hva vi kan gjøre bedre, og hvilke tiltak vi kan gjøre i neste sprint*. Denne øvelsen har blitt gjennomført i fellesskap ved å bruke den digitale samarbeidsplattformen Miro. Miro-brettet er delt inn i tre fargende bokser; en for hver av refleksjonspunktene som blir gjennomgått i fellesskap. Hvert gruppe-medlem fikk så tildelt et sett med digitale post-it-lapper og 5 minutter for å skrive ned deres refleksjoner. Når de 5 minuttene hadde passert, gikk runden rundt bordet hvor hvert av medlemmene fikk dele sine refleksjoner og tanker sammen med post-it-lappene de hadde fylt ut. Prosessen ble gjentatt totalt tre ganger slik at alle de tre seksjonene og tilhørende refleksjonspunkter ble fullstendig utfylt.



Figur 2 Tretrinns retrospective i Miro

Resultatet av øvelsen er mange post-it-lapper med gode refleksjoner om hva som fungerer og hva som kan forbedres. Ved inngangen til neste sprint blir det alltid gitt en påminnelse om å gjennomgå Miro-brettet for å skape enda tydeligere bevissthet rundt refleksjonene og forbedringsområdene. Erfaringen og resultatet av denne øvelsen viser at den fungerer godt, ettersom flere av forbedringsområdene ikke dukker opp igjen ved senere sprint retrospective-møter, og i stedet blir forbedret og oppført som positive endringer og effekter. Dette merkes også i arbeidet i påfølgende sprints. Et møtereferat ville deretter bli skrevet av scrum-masteren og lagret felles og tilgjengelig for hele scrum-teamet (se vedlegg 7 for eksempel på møtereferat).

Det er verdt å nevne et annet møte som ofte gjennomføres under scrum-metodikken - *sprint review*. På samme måte som sprint retrospective, blir sprint review-møtet avholdt ved slutten av en sprint. Imidlertid har dette møtet fokus på å vise kunden hva som har skjedd i løpet av sprinten og å få tilbakemelding på produktet og arbeidet. Gruppens arbeidsmetodikk har hatt en litt annen tilnærming til dette møtet mellom scrum-teamet og oppdragsgiver- Weekly standup.

Weekly standup

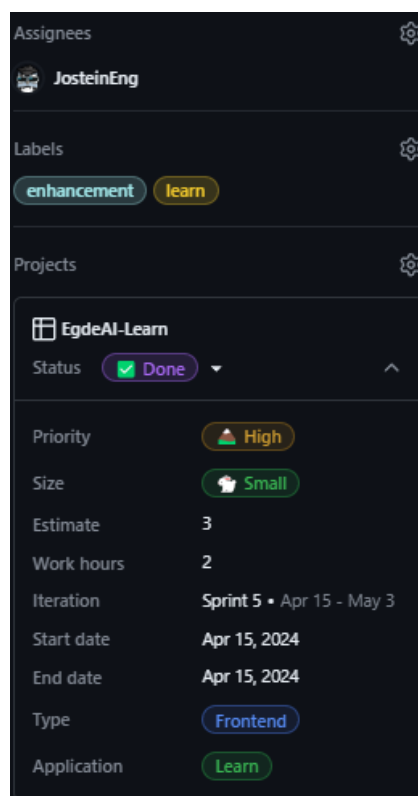
I konteksten av bachelorprosjektet har sprint review spilt en mindre rolle ved slutten av hver sprint. I stedet for å ha et fast møte ved slutten av hver sprint, har det blitt holdt ukentlig dialog med Egde, som i dette tilfellet kan anses som kunde, gjennom det gruppen har kalt ukentlige stand-up møter. Dette møtet har hatt tilsvarende effekt, men med en mye hyppigere intervall for tilbakemelding og innspill mellom scrum-teamet og Egde.

Hensikten bak disse møtene har vært å dele statusen på prosjektet, vise demoer og generelt informere om arbeidet som er blitt utført siden forrige møte. Dette har sikret åpen kommunikasjon mellom gruppen og prosjektets oppdragsgiver, men ikke minst vært en viktig arena for å måle kvaliteten opp mot Egdes forventninger og tilfredsstillelse med det utførte arbeidet, samt planene fremover.

2.5 GitHub Projects

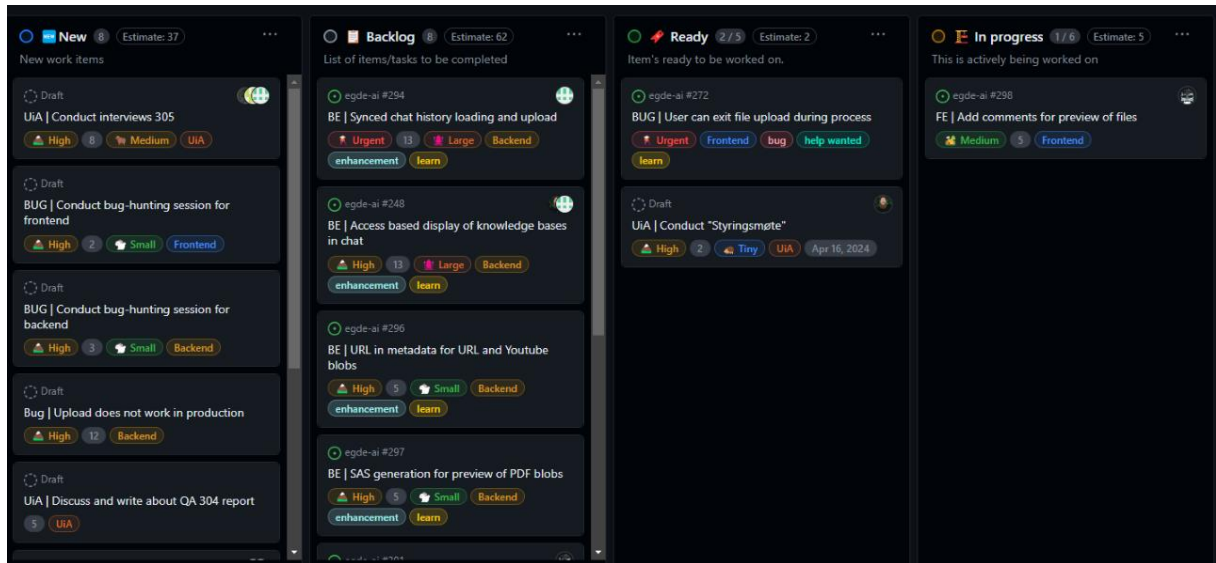
For å strukturere og bryte opp arbeidsoppgavene har de blitt loggført i GitHub sin kanban tavle løsning, GitHub projects. Her ble arbeidsoppgaver/backlogg-artikler, også kalt issues, opprettet med en beskrivende tittel, beskrivelse og ulike merknader som videre beskriver den aktuelle arbeidsoppgaven.

Dersom en arbeidsoppgave ble opprettet la man først til en tittel. Titlene på arbeidsoppgave startet enten med "UX" (design), "FE" (frontend), "BE" (backend) eller "Docs" (dokumentasjon) for å kategorisere dem. Videre ville en beskrivelse bli lagt til. Beskrivelsene hadde som hensikt å utdype hva arbeidsoppgaven handlet om, men også å loggføre progresjonen i arbeidsoppgaven. Deretter ble ulike merknader lagt til i arbeidsoppgaven. Disse merknadene var prioritet, størrelse, tidsestimat, arbeidstimer, sprint, startdato, sluttdato og type. Når en arbeidsoppgave først hadde blitt opprettet, ville den ligge i "New" frem til den ble godkjent som en arbeidsoppgave og flyttet til "Backlog". Her ble arbeidsoppgaver liggende helt til de var klare for å jobbes med, da ville de bli flyttet til "Ready".



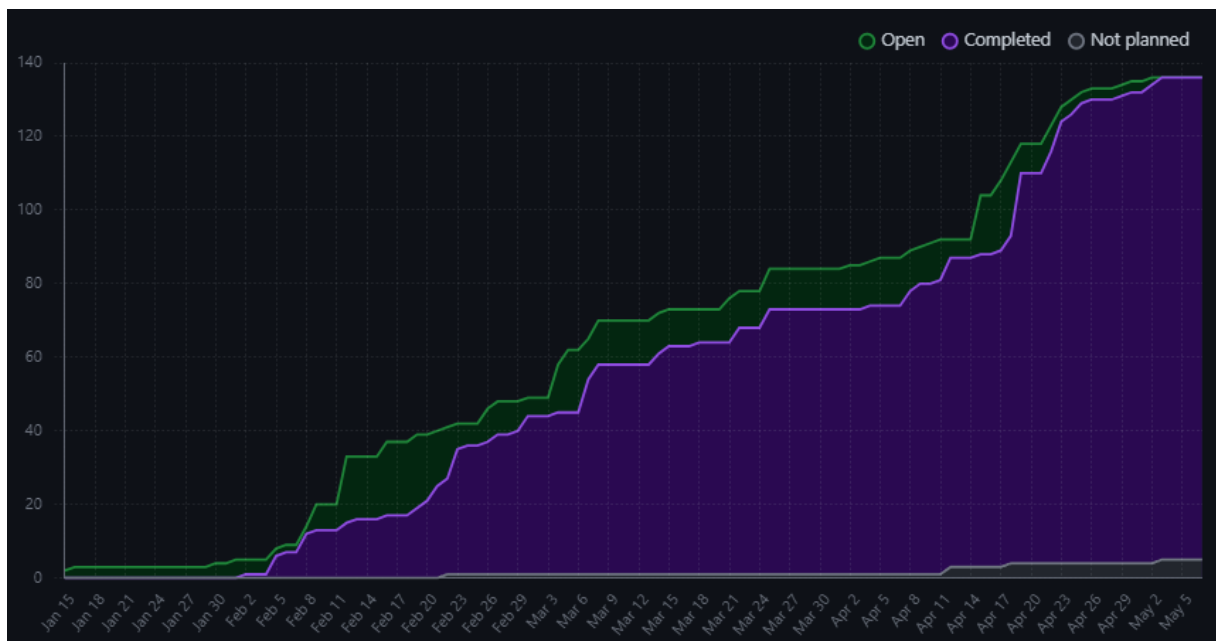
Figur 3 Skjermdump fra Github Projects

Når arbeidsoppgaver var påbegynt, ble de flyttet til "In progress" og etter hvert som de ble ferdige, til "In review". Arbeidsoppgaver som lå i "In review" måtte godkjennes som ferdige før de til slutt ble flyttet til "Done". Dersom en arbeidsoppgave ikke ble godkjent som ferdig, ville den flyttes tilbake til "In progress".



Figur 4 Skjermdump fra Github Projects

Merknadene med tidsestimatet og arbeidstimer ble ført opp i et burn up diagram, der det satte tidsestimatet ble sammenlignet med de faktiske arbeidstimene som ble brukt for å gjennomføre en arbeidsoppgave. Dette visualiserte hvor nøyaktig den totale summen av tidsestimatene var kontra de faktiske arbeidstimene som ble brukt. Akkurat hvordan disse tidsestimatene ble satt, forklares i neste delkapittel.



Figur 5 Burn up graf fra Github Projects

2.6 Estimering

Estimering av størrelsen på en arbeidsoppgave, med tanke på arbeidstimer, er noe gruppen har erfart som utfordrende fra tidligere prosjekter. Måten arbeidsoppgaver ble estimert i løpet av prosjektet var via planning poker, som går ut på at en arbeidsoppgave først blir presentert og forklart, og deretter estimert individuelt. Deretter vil alle de individuelle estimatene bli presentert samtidig. Dersom alle de individuelle estimatene er forholdsvis like, ble det endelige estimatet gjennomsnittet av alle de individuelle. Hvis det var et tilfelle der det var et stort sprik mellom estimatene, ville dette bli diskutert i fellesskap, og man ville ende opp et felles estimat man ble enig om. Om en arbeidsoppgave ble estimert med en tidsbruk i timer på tre eller flere arbeidsdager, burde den mest sannsynlig brytes ned i mindre, mer håndterbare arbeidsoppgaver.

Hver enkelt estimering blir dokumentert i backloggen, supplert med en tilhørende størrelse-tag som reflekterer den antatte tidsrammen for utførelsen av oppgaven. Estimatenes kategoriseres etter størrelse: oppgaver med estimer arbeidsomfang på 0 til 2 timer får taggen *tiny*, 3 til 5 får *small*, 6 til 8 får *medium*, 9 til 14 får *large*, og oppgaver på 16 eller flere timer får *x-large*. Dersom en oppgave er klassifisert som *x-large*, vil det vurderes om det er hensiktsmessig å dele den opp i flere mindre oppgaver.

I innledningen av prosjektløpet ble samtlige backlogg-artikler dokumentert med kun estimert arbeidsomfang. Det ble avdekket et behov innad i gruppen for å ikke bare kvantifisere det estimerte arbeidsomfanget, men også å dokumentere det faktiske tidsforbruket for hver enkelt oppgave. Dette tiltaket muliggjorde en mer omfattende oversikt over den totale tidsbruken på oppgavene, og ga grunnlag for en sammenligning med de initiale estimatene for å vurdere treffsikkerheten.

Gjennom alle de dokumenterte sprintene kunne man raskt observere et mønster med en høyere grad av overestimering sammenlignet med den faktiske brukte arbeidstiden per backlogg-artikkel. Tidlig i prosessen ble dette diskutert internt i gruppen. Gruppen konkluderte med at det var bedre å overestimere arbeidstiden for dette prosjektet for å unngå løfter om overlevering til oppdragsgiver. Eventuelle overtidsressurser kunne derfor bedre utnyttes til andre arbeidsoppgaver eller bidrag, slik at gruppen alltid kunne maksimere utnyttelsen av tilgjengelig arbeidskraft i hver sprint. Se vedlegg 9 og 10 for oversikt over estimert og registrert arbeidstid per sprint. Det er viktig å poengtere at denne bevisste overestimeringen ikke var spesielt stor. Hvis det gjennomsnittlige estimatet var mellom to antall timer, for eksempel 6.5, ville man alltid runde opp. Kort forklart, ble det heller estimert en time mer i stedet for en time mindre.

2.7 Risikoanalyse

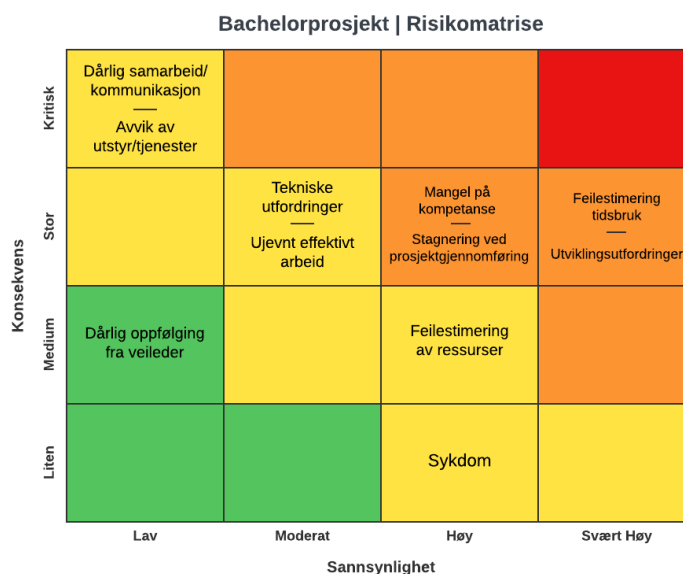
I oppstartfasen av prosjektet ble risikoanalyse satt i gang for å kartlegge potensielle barrierer i prosjektløpet. Det ble bestemt å gjennomføre to separate risikoanalyser, en for prosjektgjennomføringens del, og en for produktets del. Grunnen til dette var å kunne kategorisere hva slags område av prosjektet en risiko ville ha et utfall for, da bachelorprosjektet hovedsakelig består av utviklingen av et produkt, og hvordan man får gjennomført prosjektet.

For begge disse områdene ble risikomatriser opprettet, med konsekvens på y-aksen, og sannsynlighet på x-aksen. Innen disse ble det satt fire ulike alvorlighetsgrader. Deretter ble en rekke risikomomenter ført opp i ulike alvorlighetsgrader. Dersom et risikomoment hadde blitt satt opp på maksimum alvorlighetsgrad på både konsekvens og sannsynlighet, ville dette risikomomentet hatt maksimum innflytelse over enten prosjektgjennomføringen eller produktet.

Risikomatrisen til høyre (Figur 6) viser de ulike risikomomentene for prosjektgjennomføringen og hvordan de ble rangert med tanke på konsekvens og sannsynlighet.

Videre ble disse risikomomentenes beskrivelse og tiltak og ført opp sammen med deres respektive konsekvens, sannsynlighet og risikograd. Risikomomentene for prosjektgjennomføringen kan sees i vedlegg 11. Risikomatrisen og risikomomentenes utdyping for selve produktet ligger i vedlegg.

I påfølgende stadier av prosjektarbeidet har risikomatrisen spilt en mindre fremtredende rolle. Etter hvert som prosjektet utviklet seg, ble fokus i større grad rettet mot praktisk produktutvikling og implementering av planlagte tiltak. Som et resultat av dette skiftet fokuset bort fra den teoretiske vurderingen av risikoer, og mer mot håndgripelige oppgaver og utfordringer knyttet direkte til produktutviklingen.



Figur 6 Risikomatrise

2.8 Valg av verktøy og teknologi

Ettersom EgdeAI-Learn skulle være et produkt under den samme tekniske paraplyen som EgdeAI-Chat, falt valget av verktøy og teknologi på mye av det samme som den allerede eksisterende chat applikasjonen tok i bruk. Utenom dette ønsket gruppen å ta i bruk verktøy og teknologi som blir tatt i bruk av ansatte hos Egde, da dette ville føre til gode muligheter for veiledning fra ressursnettverket i Egde.

2.8.1 UX-verktøy

For design og prototyping av applikasjonen, ble Figma tatt i bruk. Figma er design- og prototyping-verktøyet Egde hovedsakelig tar i bruk, som førte til at gruppen hadde en del designmaler og komponenter tilgjengelig fra Egde sitt bibliotek i Figma. Det å velge Figma som verktøyet for å utføre designprosessen førte også til at designere hos Egde kunne hjelpe til med eventuelle barrierer som man potensielt kunne støte på.

2.8.2 Frontend verktøy

For frontend utvikling av applikasjonen ble React rammeverket tatt i bruk, med Tailwind for styling. Tailwind er et CSS-rammeverk som brukes for å bygge tilpassede brukergrensesnitt raskt og effektivt. I motsetning til andre CSS-rammeverk som Bootstrap eller Foundation, som kommer med forhåndsdefinerte komponenter, legger Tailwind vekt på å gi utviklere små, nedbrytbare CSS-klasser som kan kombineres på ulike måter for å bygge komplekse design. Et eksempel på dette er tailwind.config filen, som lar deg tilpasse designsystemet ditt, inkludere farger, typografi, avstand og mer, for å matche en applikasjons unike design. I tillegg til dette ble også Shadcn tatt i bruk, som er et bibliotek hvor man kan hente ned ferdige komponenter for å senere modifisere dem slik man ønsker.

2.8.3 Backend verktøy

For backend utviklingen av prosjektet har OpenAIs GPT-modeller blitt tatt i bruk for ulike funksjonaliteter. Disse modellene blir tilgjengeliggjort via en API-nøkkel. Videre ble Langchain ble tatt i bruk for å legge til mer funksjonalitet relatert til kunnskapsdatabasen som applikasjonen er koblet opp mot. Langchain er et rammeverk for utvikling av applikasjoner som operer med språkmodeller (Langchain, n.d.). I denne applikasjonen har kun JavaScript Langchain-biblioteket blitt benyttet, til tross for at Langchain tilbyr et bredt spekter av funksjonalitet.

Kunnskapsdatabasen som har blitt tatt i bruk er Pinecone, som er en databasetjeneste der man lagrer tekst i et vektorisert format. Pinecone er kunnskapsdatabasen AI-en kommuniserer med for å hente nødvendig data til å svare på spørsmål. *“En vektordatabase indekserer og lagrer vektorembeddings for rask gjenfinning og likhetssøk, med funksjoner som CRUD-operasjoner, metadatafiltrering, horisontal skalering og serverløs.”* (Oversatt fra engelsk - Schwaber-Cohen, n.d.).

For å sette opp funksjonalitet og kobling med kunnskapsdatabasen, har en rekke funksjoner fra Pinecone blitt brukt i koden. Selve koblingen med kunnskapsdatabasen har blitt gjort via en API-nøkkel. For å lagre selve filene, og for å ha en referanse til dataen som blir lastet opp i Pinecone, har Azure Blob Storage blitt tatt i bruk.

Det ble opprinnelig vurdert å implementere Azure AI Search som en løsning for søk i EgdeAI-Learn-applikasjonen. Med søk refereres det til en AI-drevet prosess som søker etter relevant informasjon for å kunne gi best mulig svar på brukerens forespørsler eller prompter. Imidlertid ble det raskt erkjent at kostnadene knyttet til Azure AI Search var uforholdsmessig høye i forhold til prosjektets budsjett. Som et resultat ble det utforsket alternativer for å oppfylle søkefunksjonaliteten til en mer overkommelig pris. Etter nøye vurdering ble Pinecone identifisert som et levedyktig alternativ, hovedsakelig grunnet dets mer kostnadseffektive natur. Det ble også tatt i betraktning at Pinecone var tatt i bruk i Egde sin eksisterende AI-applikasjon, EgdeAI-Chat.

Selve kodespråket som har blitt tatt i bruk for å utvikle både frontend og backend i applikasjonen er TypeScript, som er en utvidet versjon av JavaScript med statisk typetilordning og strengere syntaks. Applikasjonen er kodet innenfor Next.js rammeverket, som er et rammeverk for webbaserte React-applikasjoner. Next.js er utviklet av Vercel, som er plattformen applikasjonen blir kjørt og administrert fra. Vercel kobles direkte opp mot GitHub repositoret, som er der koden ligger lagret. Disse verktøyene og teknologiene blir brukt i likhet som i Chat applikasjonen. GitHub repositoret følger en mono repo struktur, der både koden for Learn applikasjonen og Chat applikasjonen ligger lagret. Tanken er at alle AI-applikasjoner fra Egde skal ligge lagret her. Det er viktig å bemerke seg applikasjonene i mono repoet blir kjørt i eget Vercel-miljø.

3. Prosjektgjennomføring

I dette kapittelet vil det beskrives de ulike stegene som er tatt i gjennomføringen av prosjektet. Det starter med en grundig kartlegging av systemkrav og brukerbehov, etterfulgt av en gjennomgang om hvordan denne innsikten har blitt omsatt til et visuelt design. Til slutt vil det fokuseres på den praktiske implementeringen av systemet.

3.1 Kartlegging av brukerbehov

I den innledende fasen av prosjektet ble det lagt betydelig vekt på UX research. For å samle relevant informasjon og få en grundig forståelse av målgruppen ble en rekke metoder tatt i bruk, herunder spørreundersøkelse, brukerintervju, utarbeidelse av personas, empatikart og brukerhistorier. Det gjennomførte arbeidet i denne fasen tjener som grunnlag for design og utvikling, forankret i konkrete brukerbehov. Den innsamlede dataen fra disse metodene utgjør en essensiell ressurs for å veilede designbeslutninger og for å sikre at sluttproduktet adresserer reelle og relevante problemstillinger for målgruppen.

3.1.1 Målgruppen

Fra starten av prosjektet har gruppen vært oppmerksom på at målgruppen for systemet vil kunne være mer enn ansatte hos Egde. Ved presentasjonen av vårt konsept for Egde, ble det tatt utgangspunkt i *Dyreparken i Kristiansand* som en illustrasjon av en typisk kunde og bruker av EgdeAI-Learn. I lys av dette har gruppen kontinuerlig hatt i bakhodet målet om å forme systemet som en tjeneste som kan tilbys ulike kundegrupper, som igjen ble bekreftet som en potensiell mulighet av prosjektets interessenter i Egde.

Starten av prosjektet handlet i stor grad om å samle informasjon om brukergruppen. For innsamling av brukerdata ble det utarbeidet et omfattende spørreskjema som ble distribuert til Egdes ansatte. Spørreskjemaet inneholdt varierte spørsmål om deres oppfatninger av AI-verktøy generelt, deres bruk av EgdeAI, samt eventuelle forslag til forbedringer eller mangler i det eksisterende systemet. Se vedlegg 12 for hele spørreskjemaet. Med over 30 deltakere ga spørreskjemaet betydelig verdifull informasjon om brukerne og oppfatningen av AI.

I spørreskjemaet ble det etterspurt av deltakere om å delta i valgfrie intervju for en mer dyptgående utforskning av deres tanker om AI. Fjorten personer svarte ja på å stille til intervju. Det var ønskelig å utføre intervjuene fysisk, og det ble da planlagt og innkalt til seks intervju. Ettersom en av intervjukandidatene meldte avvik grunnet sykdom, ble det utført fem intervjuer. Gjennom intervjuene ble det oppnådd en god innsikt i de spesifikke behovene for et slikt system. Det ble identifisert områder for forbedringer i det eksisterende systemet, og de ble identifisert nye funksjoner som var ønskelige for brukerne.

Målet med brukerintervju og innsamling av data er å kunne designe tjenesten så den møter brukernes forventinger, og ikke baserer seg på gruppens egne preferanser eller antakelser. Informasjonen innhentet fra spørreundersøkelsen og intervjuene danner et grunnlag for å forstå brukergruppen. Videre vil denne informasjonen brukes til å lage personas, empatikart, og brukerhistorier. Denne tilnærmingen styrker gruppens forpliktelse til å levere en optimalisert tjeneste designet ut fra brukernes opplevelse og behov.

3.1.2 Personas

Personas er fiktive representasjoner av en målgruppe eller brukertype (Bruton, 2022). Hensikten med å utvikle personas er å skape en felles forståelse for brukerne innad i gruppen, slik at alle kan ha et klart bilde over hvem sluttbrukerne er. Personas fungerer som et nyttig verktøy gjennom hele utviklingsprosessen som bidrar til å holde fokus på sluttbrukeren og ta beslutninger i tråd med brukernes behov. Til dette prosjektet ble det utviklet fire personaer, som representerer ulike aspekter av målgruppen. Se vedlegg 13 for de resterende personaene.

Biografi

- Elisabeth er en kreativ og lidenskapelig UX-designer. Etter å ha fullført sin utdanning innen tjenstedesign, og med 3 års erfaring i konsulentbransjen, har hun etablert seg som en talentfull profesjonell innen brukeropplevelsesdesign.

Holdning

- Aktiv og energisk person som ikke liker å sitte stille.
- Liker å ta i bruk mennesker rundt seg for å finne hjelp til ting hun lurer på.

Mål

- Ønsker å forbedre brukeropplevelser gjennom innovative og brukervennlige designløsninger.
- Kontinuerlig utvikle sine ferdigheter og holde seg oppdatert på nye designretninger og teknologier.

Erfaring

- Har hørt at AI-verktøy kan bidra til mer effektivt arbeid, men har valgt å ikke ta det i bruk i eget arbeid grunnet manglende tillit til AI.

Elisabeth Grønnskog
 Alder: 26 år - f. 1998
 Utdanning: Tjeneste design
 Rolle: UX design
 Ansenittet i bedriften: 3 år
 Erfaring: 3 år

Figur 7 Persona Elisabeth Grønnskog

Biografi

- Mads er en erfaren og kunnskapsrik fullstack-utvikler. Han har på kort tid blitt en viktig del av teamet sitt ved å bringe sin ekspertise innen både frontend og backend utvikling.

Holdning

- Strukturert og rutinebasert person som liker å holde seg oppdatert.
- Verdsatter gode sosiale forhold med sine medarbeidere.

Mål

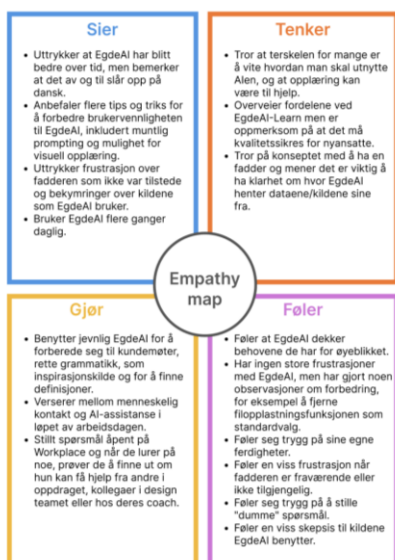
- Ønsker å være en verdifull bidragsyter til teamet, og levere prosjekter med høy kvalitet innen tidsfristene.
- Har ambisjoner om å være involvert i prosjekter som utfordrer han og gir rom for kreativ problemløsning.

Erfaring

- Tar aktivt i bruk og tester nye AI-verktøy i søket om å bli mer effektiv i sitt arbeid.

Mads Mikkelsen
 Alder: 32 år - f. 1992
 Utdanning: Informatikk
 Rolle: Utvikler
 Ansenittet i bedriften: 1/2 år
 Erfaring: 9 år

Figur 8 Persona Mads Mikkelsen



Figur 9 Empatikart

3.1.3 Empatikart

For å få oversikt over informasjonen samlet inn fra spørreundersøkelsen og brukerintervju, ble det utviklet empatikart. Disse visualiseringene representerer hva en bruker tenker, sier, gjør og føler (Figma, u.å.). Dette kan hjelpe gruppen å legge mer vekt på brukernes perspektiv og styrke empatien for brukergruppen. Det ble utviklet flere empatikart (se vedlegg 14), som ble brukt til å lage et felles kart som oppsummerer tema og mønster fra brukerne som kan ses i Figur 9.

3.1.4 Brukerhistorier

Brukerhistorier er korte, enkle setninger som fokuserer på brukernes behov (Coursera, 2023). Det ble utviklet fem brukerhistorier til dette prosjektet, som representerer fem mulige brukere; nyansatt, prosjektleder, UX-designer, utvikler, og regnskapsfører. Disse hjelper gruppen å holde oversikt over hvem, hva og hvorfor (hvem er brukeren? hva trenger de? hvorfor vil de det?). Videre ble tre av disse visualisert for å få oversikt over reisen brukeren må gjennom for å oppnå målet sitt. Se vedlegg 15 for resterende brukerhistorier.

As a [type of user],
I want [an action]
so that [a benefit/a value].

- som en nyansatt, ønsker jeg hjelp til å lære ulike verktøy så jeg kan bidra i prosjektteamet mitt.
- som en prosjektleder, ønsker jeg lett tilgang på informasjon i EgdeNett så jeg kan spare verdifull tid.
- som en ux-designer, ønsker jeg å få inspirasjon og øke kreativiteten så jeg kan komme raskt i gang med arbeid.
- som en utvikler, ønsker jeg hjelp med kode så jeg slipper å bruke tid på bug-fixing.
- som en regnskapsfører, ønsker jeg å hente informasjon om bedriften så jeg ikke trenger å søke opp manuelt.



Figur 10 Brukerhistorier

Figur 11 Visuell brukerhistorie av nyansatt

3.2 Kartlegging av systemkrav

For å kartlegge systemkrav vil det først gis en systemdefinisjon hvor de primære funksjonene og elementene ved systemet forklares. Videre vil det gis en gjennomgang av systemkriterier og en prioritering av disse.

3.2.1 Systemdefinisjon

En systemdefinisjon utgjør beskrivelsen av et system og forklarer de primære egenskapene og elementene ved produktet.

EgdeAI-Learn har som hovedoppgave å kommunisere tekst i et samtaleformat med brukere, basert på et datagrunnlag definert av administratorer. For å oppnå dette måtte en rekke funksjonaliteter implementeres. Administratorer måtte ha tilgang til en side der de kunne administrere kunnskapsbasene og tilhørende data. På brukersiden måtte det være mulig å starte samtaler med AI-en basert på kunnskapsbasene og interagere med dataen som er tilgjengelig i den aktuelle kunnskapsbasen. Denne funksjonaliteten, sammen med en rekke andre funksjoner, definerer hva slags system EgdeAI-Learn er.

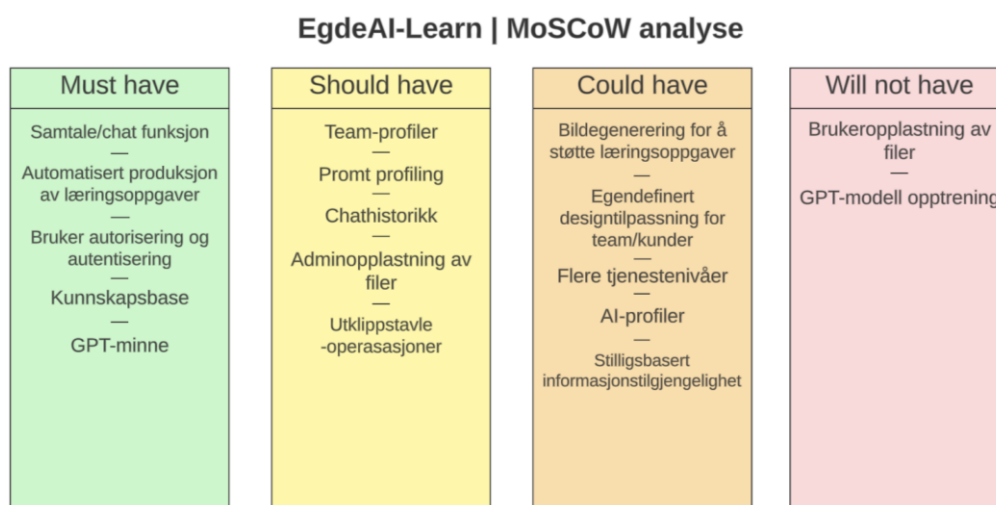
Ved prosjektstart ble systemets definisjon utarbeidet ved bruk av FACTOR-modellen. Dette er et hjelpemiddel som Mathiassen omtaler (2018, s. 24) ved prosessen når man skal formulere en systemdefinisjon. For dette prosjektet ble FACTOR-modellen tatt i bruk hovedsakelig i starten for å utarbeide et felles overblikk over tjenesten. Her ble det definert funksjonalitet, applikasjonsdomene, forhold, teknologi og objekt. Siden oppgaven handlet om utforskning og smidig utvikling, var ikke systemdefinisjonen et fastsatt konsept, men åpen for endringer. Modellen ble derfor brukt som et hjelpemiddel i starten for å ha et utgangspunkt som ville utforskes videre. Se vedlegg 17 for den ferdig utfylte FACTOR-modellen for systemet EgdeAI-Learn.

3.2.2 MoSCoW

MoSCoW-analyse er et verktøy brukt for å få oversikt over de ulike funksjonene i systemet, samt hvordan disse skal prioriteres under utviklingsprosessen. MoSCoW er et akronym som står for de fire ulike kategoriene funksjonalitetene kan bli delt opp i. Kategoriene er: *must have*, *should have*, *could have* og *will not have* (VanZandt, 2023).

Must have inkluderer de funksjonalitetene som er essensielle for systemet. *Should have* omfatter de funksjonalitetene som er viktige, og bør implementeres dersom det er mulig. De er ikke essensielle, men bringer stor verdi for produktet. *Could have* inkluderer de funksjonalitetene som er fine å ha med, men kan vurderes basert på tid og ressurser om de lar seg implementeres. *Will not have* vil inkludere funksjonaliteter som ikke vil bli prioritert eller implementert i systemet. Dette kan innebære funksjonaliteter som er utenfor omfanget av dette prosjektløpet, men som kan vurderes for fremtidige iterasjoner.

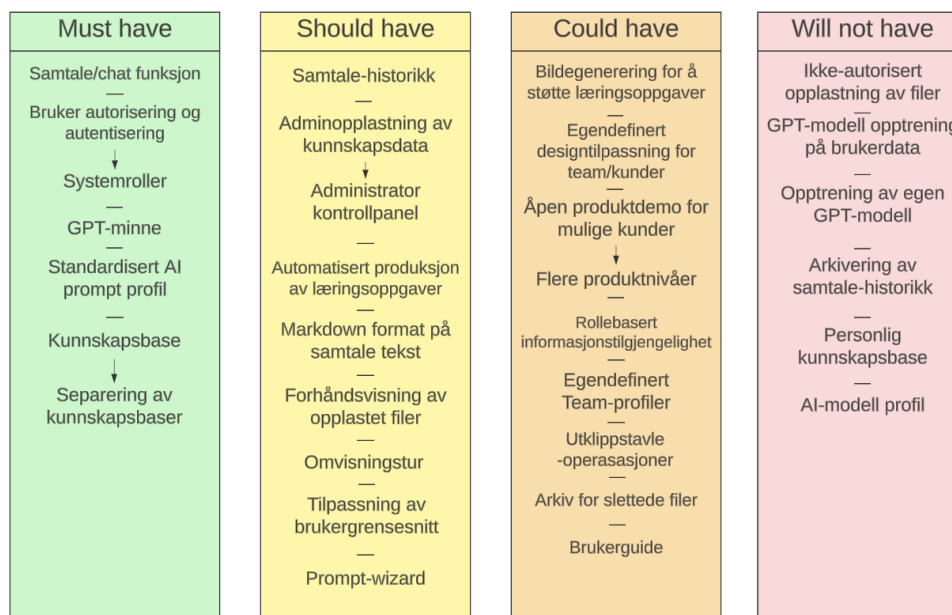
I innledningen av prosjektløpet ble det gjennomført en MoSCoW-analyse for å systematisere de ulike funksjonalitetene som skulle inkorporeres i systemet. Dette verktøyet ble anvendt for å skape klarhet rundt hvilke funksjoner som var av avgjørende betydning, viktige, ønskelige, og hvilke som var mindre kritiske for prosjektets suksess. Gjennom en kollektiv diskusjon ble gruppen enige om denne funksjonalitetsklassifiseringen, basert på samlet innsikt og prioriteringer.



Figur 12 Original MoSCoW analyse

På grunn av prosjektets åpne karakter og usikkerheten som fulgte med, angående både produktets endelige form og potensielle utfordringer som kunne oppstå, ble det identifisert behovet for å revidere denne analysen underveis. I Sprint 4 ble det derfor holdt et møte med alle gruppemedlemmene til stede, hvor formålet var å gjennomgå og oppdatere vurderingen av de ulike funksjonalitetene i systemet, samt legge til nye og oppdatere de eksisterende funksjonalitetene.

EgdeAI-Learn | MoSCoW analyse



Figur 13 Oppdatert MoSCoW analyse

Analysen av MoSCoW har i dette prosjektet fungert som et verktøy for å sikre konsensus blant gruppemedlemmene angående prioriteringen av systemfunksjonalitet. Ved å etablere en felles forståelse for hvilke funksjoner som er av høyest verdi og dermed bør prioriteres, har denne tilnærmingen bidratt til å skape en tydelig ramme for utviklingsarbeidet. Denne forståelsen har igjen lagt grunnlaget for en effektiv utvikling av sprint backlogg ved start av nytt sprint.

MoSCoW-analysen ble dermed ikke direkte benyttet som et verktøy for å velge funksjonaliteter som skulle inkluderes i hver sprint. Sprintprioriteringer ble bestemt internt i teamet gjennom diskusjoner og konsensus. MoSCoW-analysen fungerte derfor som et referansepunkt og en felles forståelse av produktets overordnede mål og prioriteringer, i stedet for å være en direkte operativ guide i sprintplanleggingen. Denne tilnærmingen tillot gruppen å tilpasse seg endringer og utfordringer underveis i prosjektet, mens de samtidig holdt seg tro mot de overordnede målene som ble identifisert gjennom MoSCoW-analysen.

3.2.3 Prioritering av systemkriterier

Systemkriterier utgjør en rekke retningslinjer som kan veilede til å definere viktigheten av hvert krav i systemet. Disse kriteriene kan veilede til å tilpasse systemarkitekturen som er i samsvar med produktets kontekst. De har hjulpet gruppen til å prioritere produktegenskapene og legger grunnlaget for designvalgene. Systemkriteriene har spilt viktig rolle for hvordan systemet skulle struktureres og bygget opp. Gruppen har tatt spesielt hensyn til kriteriene usability og comprehensibility, som vektlegges også av Mathiassen i "Object Oriented Analysis & Design" (2018, s. 181). Disse to kriteriene sikrer at systemet er brukervennlig og lett å forstå. Figur 14 viser prioriteringen av de ulike kriteriene. Forklaring av de ulike kriteriene kan ses i vedlegg 19.

Kriterie	Veldig viktig	Viktig	Mindre viktig	Irrelevant
Brukervennlig (Usable)	x			
Sikker (Secure)	x			
Effektiv (Efficient)		x		
Korrekt (Correct)	x			
Troverdig (Reliable)	x			
Vedlikeholdbar (Maintainable)		x		
Testbar (Testable)			x	
Fleksibel (Flexible)		x		
Forståelig (Comprehensible)	x			
Gjenbrukbar (Reusable)		x		
Flyttbar (Portable)			x	
Samvirkosom (Interoperable)			x	

Figur 14 Prioritering av systemkriterier

3.3 Beslutninger som formet prosjektet

Gjennom prosjektløpet er det blitt tatt flere viktige beslutninger som har formet prosjektet. Dette inkluderte endringer i brukergrensesnittet for å forbedre brukeropplevelsen, implementeringen av en administratorside for dynamisk innholdshåndtering og avgjørelsen om å avstå fra arkiveringsmuligheter for chat-historikk basert på brukerens behov og interesse. Disse valgene ble gjennomført grunnet nye behov som oppstod fra brukeren og arbeidsgiveren på forskjellige tidspunkt gjennom prosjektløpet.

3.3.1 Tidlig designendringer

Tidlig i designprosessen ble det gjennomført en workshopsesjon med en erfaren UX-designer fra Egde. Han fremmet innsiktsfulle forslag som fokuserte på å forbedre brukeropplevelsen ved å gi en mer innbydende "chat-følelse" ved å raskt dirigere brukerne til hovedsiden. Han argumenterte også for at fleste brukerne har preferanse for å direkte starte samtaler med AI-en, derfor ble det besluttet etter møtet å omgjøre på designet, slik at brukerne fikk direkte tilgang til chatfunksjonen. Det opprinnelige designutkastet ble ansett som tungvint, da den først dirigerte brukerne til området med knapper som representerer kunnskapsbaser, før de ble omdirigert til selve chatsiden. Dette møtet førte til betydelige endringer i designet av «landingssiden». Mer om dette møtet og prosessen som førte til disse endringene kan leses i delkapittel 4.2.

3.3.2 Administratorside

Implementeringen av en administratorside var et avgjørende valg som påvirket prosjektets retning og funksjonalitet. Et spørsmål som oppsto tidlig i prosjektløpet var knyttet til hvem som skulle ha tillatelse til å laste opp data og informasjon til kunnskapsbasene. Etter samtaler både med arbeidsgiver og internt i gruppen ble det besluttet om å implementere en administratorside. Oppdragsgiveres interesse for å gi både kunder og Egde muligheten til å laste opp materiale, veide tungt i avgjørelsen.

Avgjørelsen om implementeringen av administratorsiden ble tatt senere i designfasen og hadde en betydelig innvirkning på prosjektet. Dette resulterte i vesentlige endringer, ettersom en betydelig mengde arbeidsressurser og arbeidstid ble allokert til implementeringen av denne siden. Det var nødvendig å utvikle en brukervennlig og sikker plattform for å opplaste innhold fra PDF-er og URL-er til kunnskapsbasen. Dette involverte både design, utvikling og testing for å sikre at funksjonaliteten var pålitelig for brukerne.

Kunnskapseiere ble gitt rettigheter til å administrere innholdet i kunnskapsbasen. Ved å ha muligheten til å opprette nye kunnskapsbaser og laste opp PDF-filer eller URL-er til nettsider direkte fra administrator grensesnittet, fikk de full kontroll over innhold og oppdatering av kunnskapsbasen. Dette tilrettela for rask oppdatering og utvidelse av responsen til EgdeAI-Learn ved nye informasjonsbehov. Administratorsiden har lagt til rette for å ha en mer dynamisk og oppdatert kunnskapsbase som kan tilpasses i takt med endrede krav og ny informasjonstilgang.

Etter kort tid besluttet utvikler teamet hos EgdeAI-Chat å selv implementere en slik administratorside. Ved demo visning av hvor effektivt brukeren kunne ta i bruk administratorsiden og laste opp informasjon, samt lage nye kunnskapsbaser ble det ønsket et slikt system hos EgdeAI-Chat. Dette bekrefter at avgjørelsen og det ekstra arbeidet med å implementere et slikt brukervennlig system var nyttig.

3.3.3 Arkivering av chat-historikk

I løpet av prosjektgjennomføringen ble det foretatt en grundig analyse av ulike implementeringsalternativer. Et vesentlig valg som ble vurdert, var implementeringen av arkiveringsfunksjonalitet for chat-historikk. Etter en grundig vurdering basert på innsikt fra brukerintervjuer og brukertester, besluttet gruppen å avstå fra denne implementeringen.

Analysen av innsamlede brukerdata avdekket en generell mangel på interesse og lav opplevd verdi blant brukerne når det gjaldt arkiveringsfunksjonaliteten. Videre evaluering av den innsamlede informasjonen indikerte tydelig at brukerne ikke anså det som ønskelig å ha muligheten til å arkivere chat-historikk. De ble spesifikt spurt om deres praksis med å organisere chat-historikk ved bruk av lignende verktøy, og flertallet svarte enten at de ikke gjorde noe særlig med den, eller at de slettet samtaler etter behov.

Selv om mange av brukerne hadde erfaring med lignende verktøy som tilbød arkiveringsfunksjonalitet, uttrykte de likevel at de ikke så behovet for dette i denne konteksten. Dermed ble det bestemt at det var mest hensiktsmessig å avstå fra å integrere en arkiveringsfunksjon for chat-historikk, og heller allokere ressursene og fokuset over til funksjonaliteter som ville gi mer verdi til brukerne.

3.4 Design av brukergrensesnitt

Etter at gruppen hadde samlet informasjon og oppnådd en felles forståelse av målgruppen kunne designprosessen starte. I løpet av denne prosessen ble det holdt flere møter med erfarne UX-designere fra Egde. Gruppen ble introdusert til bedriftens designmanual og mottok relevant innspill angående designretninger og ønsker fra bedriften. Det ble gitt en introduksjon til komponentbiblioteket. Gruppen hadde også mulighet til å presentere wireframes og prototyper for designerne og få tilbakemelding og veiledning i løpet av designprosessen.

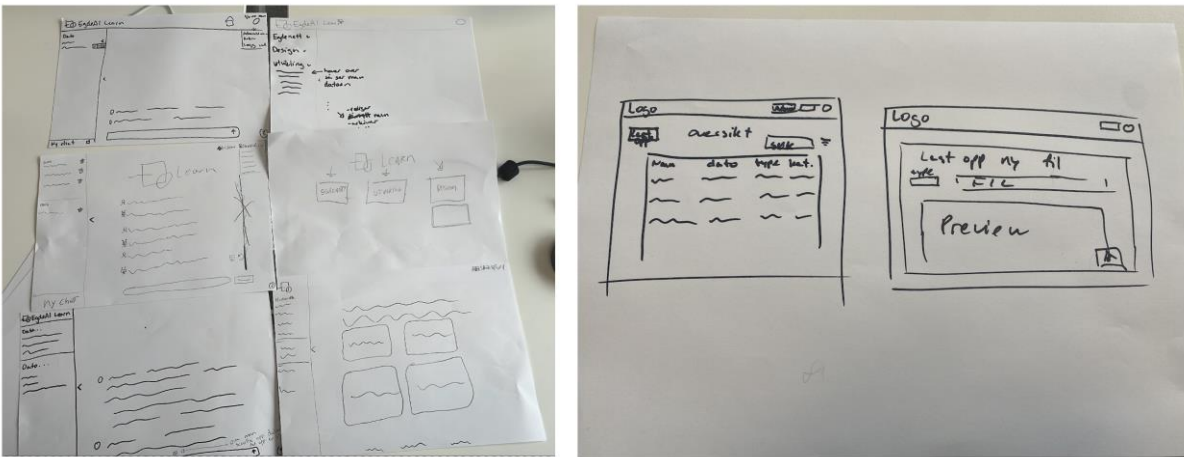
3.4.1 Designinspirasjon

Gruppen ble som nevnt introdusert til Egdes designmanual og komponentbibliotek i Figma. Utenom ønsket fargepalett og komponentbruk sto gruppen fri til å designe applikasjonen selv. I løpet av både brukerintervju og samtaler med designere kom det frem ønsker angående brukergrensesnittet som gruppen tok i betraktning under designprosessen. Gruppen ble blant annet oppfordret til å ta inspirasjon fra Egdes hjemmeside. Overordnet sett hentet gruppen inspirasjon fra Egdes merkevareidentitet i utformingen av tjenesten. Målet var å utvikle en applikasjon som harmoniserte med bedriftens etablerte merkevare.

3.4.2 Skissering

Med en felles forståelse for brukerne og med inspirasjon fra Egde kunne gruppen komme i gang med å utforske ideer til brukergrensesnittet. Skissering spiller en essensiell rolle i designprosessen, og gir rom for å utforske løsninger og muligheter for tjenesten (Rojas, 2023). Skissering skjer tidlig i designprosessen, men er en vedvarende prosess som ble utført på flere stadier i løpet av prosjektet. Samtlige gruppemedlemmer deltok i denne prosessen, enten ved å lage skisser eller ved å diskutere ideene fra skissene i lag. Dette var viktig for å sikre at designet kunne utføres og programmeres visuelt, men også at funksjonene designet la opp til var mulig å implementere.

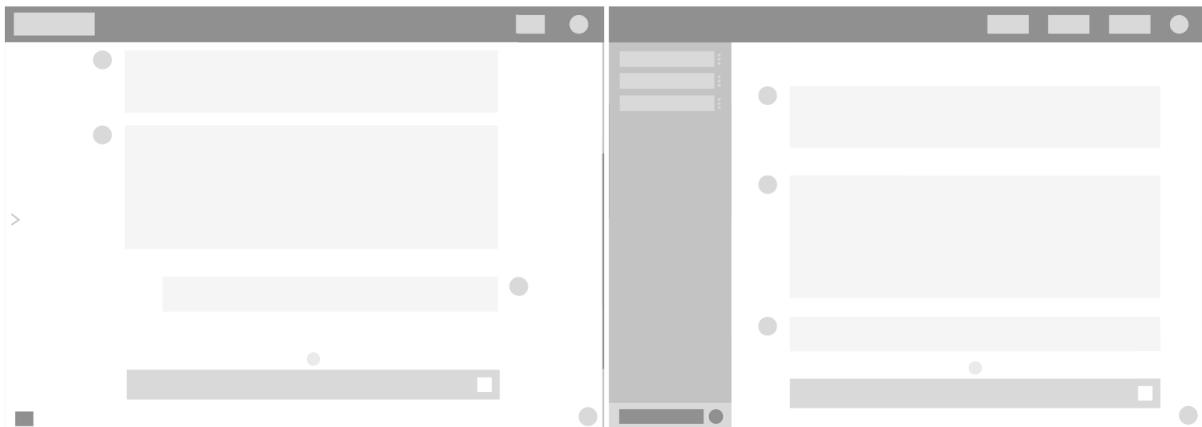
Idemyldringen og kommunikasjonen som ligger i skisseringsprosessen er essensiell for å finne de beste løsningene, og tilrettelegge for at designprosessen videre går effektivt for seg. Ved å ikke legge grundig arbeid i denne prosessen kunne gruppen risikert å måtte gjøre store endringer senere i prosjektløpet, eller ende opp med en applikasjon som ikke tilfredstilte brukerbehov.



Figur 15 Skisser av chat og administrator-siden.

3.4.3 Wireframes

Wireframes er en visuell representasjon av applikasjonen som gir et overordnet bilde av layouten og strukturen til det endelige produktet (Figma, u.å.). Formålet med wireframes er å presentere de grunnleggende elementene uten å fokusere på visuelle detaljer som farger, bilder eller skrifttyper. I dette prosjektet ble det utviklet både low fidelity (lowfi) wireframes og high fidelity (highfi) wireframes. Forskjellen mellom lowfi og highfi ligger i graden av kompleksitet og detaljer. I denne prosessen hadde gruppen mulighet til å eksperimentere med ideene fra skissene, og begynne å forme det endelige utseendet.



Figur 16 Low Fidelity Wireframe av chat i Figma.

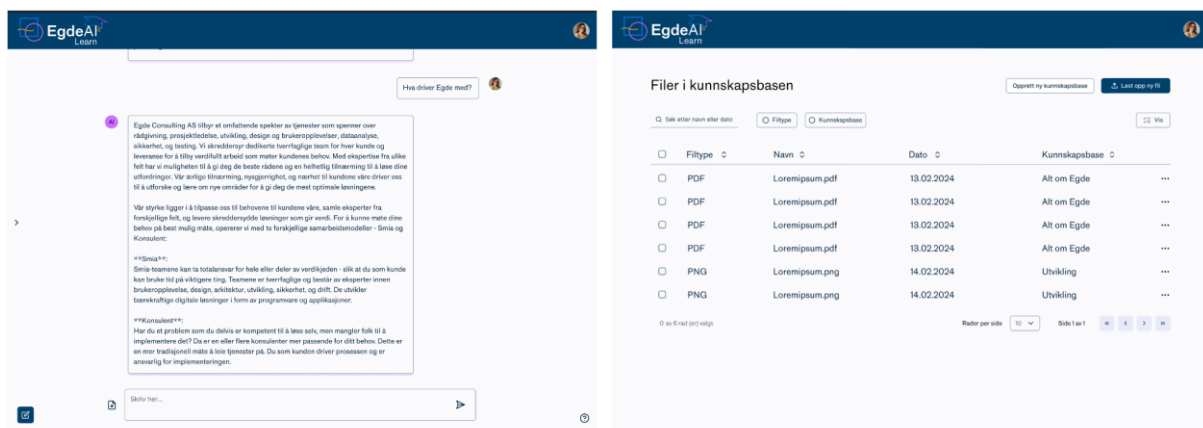
Etter å ha kommet i enighet om den grunnleggende layouten, startet prosessen med å utvikle highfi wireframes. Disse er mer detaljrike og nærmer seg mer det endelige utseende av applikasjonen. Her inkluderes også grafiske elementer som farger, bilder og skrifttyper (Figma, u.å.). I dette prosjektet har gruppen som nevnt tatt i bruk komponentsamlingen *Shadcn*, som også ligger tilgjengelig som en Figma-fil. Denne ble tatt i bruk for å lage highfi wireframes og prototype, som minsker mengden arbeid med å designe komponenter selv, og designet blir mer likt det som etter hvert vil lages i front-end. Dette vil videre føre til mer verdi i forhold til brukertester. Ved å ha en prototype som er mest mulig lik det endelige designet vil det kunne testes tidlig om brukerne finner det brukervennlig.

3.4.4 Prototype / mockup

En prototype er en tidlig versjon av en løsning eller produkt som blir skapt for å teste og evaluere designkonsepter (Figma, u.å.). Formålet med å utvikle prototyper er å visualisere og teste designideene før implementering. Dette stadiet er en videreføring av wireframe og gir muligheten til å teste brukerinteraksjoner, identifisere potensielle problemer, og samle verdifull tilbakemelding fra både brukere og interessenter. Gjennom iterativ utvikling og testing åpner en prototype mulighetene for å finpusse designet og sikre at sluttproduktet oppfyller brukerbehov på en optimal måte.

Prototypen ble utviklet i Figma, og det ble utviklet prototype for både desktop, mobil og administrator-siden. Visuelt sett ligner de på highfi wireframes, men prototypene har interaksjonsmuligheter som simulerer hvordan sluttproduktet vil fungere. Dette er viktig for å samle mest mulig informasjon og tilbakemeldinger fra brukertestene.

(Lenke til prototypene ligger i vedlegg 23 og 24)



Figur 17 Prototype av chat og administrator-siden i Figma.

3.4.5 Grafiske designprinsipper og retningslinjer

Designprinsipper og retningslinjer fungerer som et rammeverk som guider designet gjennom hele prosessen, fra konseptualisering til implementering, og bidrar til å skape brukervennlige designløsninger. Til dette prosjektet har det blitt valgt å ta i bruk en kombinasjon av Benyons designprinsipper og Web Content Accessibility Guidelines (WCAG), som legger et helhetlig fundament for høykvalitets digitale opplevelser.

Benyons designkriterier

For å sikre at brukergrensesnittet er brukervennlig og behagelig for brukerne har gruppen valgt å vurdere designet basert på Benyons 12 design prinsipper (se vedlegg 25 for definisjon av designprinsippene). I vurderingen vil de bli vist eksempler fra desktop, mobil, og administrator-siden. Først vil det bli gitt en kort forklaring på hva de ulike prinsippene går ut på. Videre vil det forklares hvordan gruppen har tatt i bruk prinsippene for å evaluere designgrensesnittet, samt vises hvordan gruppen har prioritert de ulike prinsippene etter relevans for produktet.

Benyons 12 design prinsipper er delt inn i tre hovedkategorier; *learnability*, *effectiveness*, og *accommodation*. Learnability handler om hvor enkelt det er for brukerne å lære seg systemet og hvor intuitivt det føles uten noen forkunnskaper. De første 4 prinsippene går under denne kategorien. Neste kategori er effectiveness, og inkluderer prinsipper 5-9. Disse handler om å gjøre applikasjonen effektiv i bruk, hvor enkel den er å navigere seg gjennom og om brukerne føler en form for kontroll. Den siste kategorien er accommodation og handler om tilrettelegging for brukerne, som innebærer å gi brukerne valgmuligheter, fint oppsett og passende farger, og at applikasjonen oppleves som vennlig. Prinsipper 10-14 går under denne kategorien. Disse prinsippene vil bli vurdert og gitt en prioritering fra *veldig viktig*, *viktig*, *mindre viktig* eller *irrelevant* (Figur 18).

Videre vil det gis en score fra 1-100 som forteller hvor godt systemet svarer eller tilrettelegger for hvert prinsipper. I vedlegg 26 vises det konkrete eksempler på hvordan designet svarer på de ulike designprinsippene som er vurdert som veldig viktig eller viktig (visibility, consistency, familiarity, affordance, navigation, feedback, constraints, flexibility, style, og conviviality).

Vurdering av score

Vurderingen av score var et strategisk valg tatt av gruppen for å sikre at systemet tilfredsstiller de ulike prinsippene tilstrekkelig. Gruppen valgte å gjennomføre to gjennomganger og vurderinger av score. Dette ble gjort for å tillate løpende vurdering og forbedring gjennom utviklingsprosessen. Den første gjennomgangen fant sted midtveis i sprint 4, på et tidspunkt hvor de fleste elementene og funksjonalitetene hadde begynt å ta form. Å gjennomføre en vurdering av designet basert på prinsippene på dette stadiet ga gruppen en omfattende oversikt over områder som hadde behov for justeringer.

Det ble avholdt møte hvor alle gruppemedlemmene var til stede, der hvert designprinsipper ble gjennomgått. For hvert prinsipper ble det først gitt en definisjon av prinsippet, etterfulgt av en felles diskusjon om hvordan systemet presterer. Deretter ga hvert gruppemedlem sin score fra 1 til 100. For å beregne en samlet score ble gjennomsnittet regnet ut.

Under den første vurderingen av score ble det identifisert flere mangler, spesielt i forhold til tilbakemelding og begrensninger på administrator-siden. Disse funnene understreket områder som hadde behov for finjusteringer.

Kriterer	Veldig viktig	Viktig	Mindre viktig	Irrelevant	Analyse score	Ny analyse score
Synlighet (Visibility)	x				87	94
Konsistens (Consistency)		x			81	87
Igjenkjennskap (Familiarity)		x			83	91
Tydighet (Affordance)	x				91	92
Navigasjon (Navigation)		x			73	90
Kontroll (Control)			x		88	92
Tilbakemelding (Feedback)	x				59	91
Gjenoppretting (Recovery)			x		68	*68
Begrensninger (Constraints)	x				47	83
Fleksibilitet (Flexibility)		x			80	*80
Stil (Style)	x				92	95
Vennlighet (Conviviality)		x			87	*87

Figur 18 Benyons designprinsipper med score.

Web Content Accessibility Guidelines (WCAG)

Web Content Accessibility Guidelines er et sett med retningslinjer utviklet av Web Accessibility Initiative under World Wide Web Consortium for å forbedre tilgjengeligheten av netttinnhold for mennesker med ulike funksjonshemninger (Henry, 2024). Ettersom tjenesten som utvikles muligens skal selges videre til kunder av bedriften, er det ikke helt klart hvem mulige sluttbrukere kan bli. Ved å ta i bruk WCAG sikrer gruppen at tjenesten er tilgjengelig for personer med ulike funksjonshemninger. Men, resultatet av å oppfylle WCAG-kravene resulterer ofte også i en forbedret brukervennlighet for alle, uavhengig av funksjonsevne. For å oppfylle disse kravene kreves det samarbeid mellom både designere og utviklere, og det presenteres klare standarder for å sikre brukervennlighet og tilgjengelighet.

Evalueringen er av versjon WCAG 2.2, conformance level AA. Den består av 55 kriterier, fordelt under fire kategorier; *perceivable*, *operable*, *understandable*, og *robust*. Det ble gjort en manuell gjennomgang for å evaluere siden, men det ble også tatt i bruk verktøyet WAVE. WAVE er utviklet av WebAIM.org og brukes for å teste tilgjengelighet og identifisere feil eller mangler i samsvar med WCAG. Under vil det bli gitt en overordnet gjennomgang av resultatet. Se vedlegg 27 for den fulle rapporten.

Summary

Reported on 55 of 55 WCAG 2.2 AA Success Criteria.

- 38 Passed
- 2 Failed
- 0 Cannot tell
- 15 Not present
- 0 Not checked

Figur 19 Oversikt av resultat WCAG rapport

Grunnet prioriteringer og tidsbegrensninger i prosjektløpet var det mest hensiktsmessig å utføre WCAG-rapporten litt før hele systemet var ferdig utviklet. Det hadde vært gunstig å utføre rapporten etter at alle funksjonaliteter var implementert for å sikre at produktet i sin endelige form ble grundig vurdert, likevel opplevde gruppen det som svært nyttig å evaluere systemet underveis. Som følge av dette kan det være mangler eller variasjoner fra rapporten og det endelige sluttproduktet. Likevel ga rapporten en viktig gjennomgang av systemet og la opp for mer tilrettelagt og brukervennlig utvikling fremover i prosjektløpet.

Noen av kriteriene er irrelevante for EgdeAI-Learn. For eksempel de 5 kriteriene som går under "1.2 Time-based Media" handler om forhåndsinnspilt media som video og audio, noe tjenesten ikke inneholder. 2 kriterier er markert som "failed". Det ene kriteriet går ut på at innhold ikke begrenses til en enkelt visningsretning, for eksempel stående eller liggende. På mobil er applikasjonen delvis funksjonell liggende, men mister litt innhold og funksjoner når sidepanel åpnes.

Det andre kriteriet handler om å gjøre brukere som tar i bruk assisterende teknologi som skjermleser oppmerksom på endringer i innhold. Skjermleser fungerer på nettsiden, men ved interaksjoner som for eksempel å bytte utseende fra mørk til lys, vil ikke skjermleseren fortelle at det ble gjort en endring. Basert på andre prioriteringer, tidsrammene for dette prosjektet, og lav sannsynlighet for store negative konsekvenser av å ikke overholde disse kravene, vil disse kriteriene nedprioriteres i utviklingen.

3.5 Implementering av systemet (Front- og Backend)

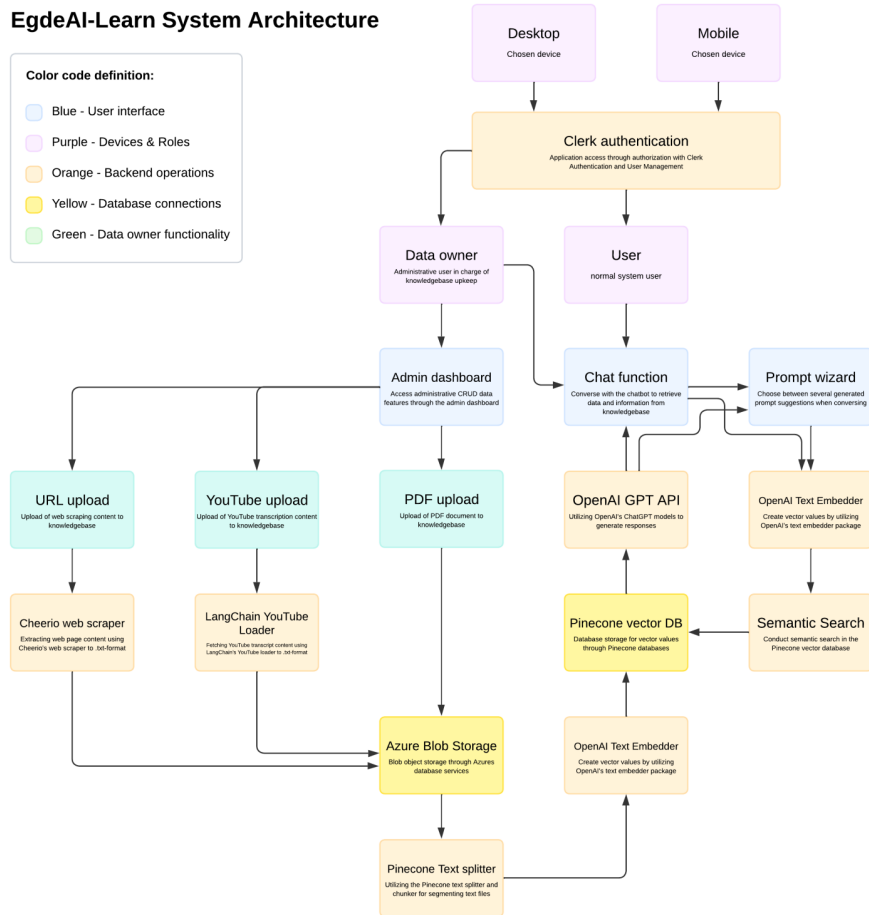
Applikasjonen er først og fremst designet for enkelhet og effektivitet, men under panseret opererer EgdeAI-Learn med en rekke sofistikerte funksjoner som skaper den sømløse brukeropplevelsen for både ordinære brukere og kunnskapseiere (administratorer).

I dette prosjektet har gruppen dykket ned i flere nøkkelområder av systemutvikling for å gi liv til EgdeAI-Learn. Senere i dette delkapitlet utforskes den komplekse systemarkitekturen som danner grunnlaget for applikasjonen, samt detaljene om implementeringen på både frontend og backend. Det blir også sett nærmere på utfordringene som oppstod underveis og de sikkerhetstiltakene som ble iverksatt for å sikre en brukervennlig og trygg applikasjon.

3.5.1 Systemarkitektur

For å mer effektivt illustrere den potensielt innviklede sammenhengen mellom et systems oppgaver og funksjoner, kan man bruke en billedlig fremstilling av systemarkitekturen for å vise sammensetningen og koblingene i systemet. Den kjente *ArchiMate*-modellen blir ofte brukt til å beskrive og kommunisere systemarkitekturen til et gitt system, applikasjon og lignende blant mange bedrifter i IT-landskapet (LeanIX, u.å.).

EgdeAI-Learn består av flere sentrale noder og knutepunkter som spiller avgjørende roller rundt applikasjonens funksjonalitet og struktur. Å beskrive denne komplekse sammensetningen kun med tekst kan være utfordrende. Derfor kan en visuell fremstilling, som for eksempel en *ArchiMate*-modell, være en mer effektiv måte å forklare systemets struktur på. Systemarkitekturen til EgdeAI-Learn tar inspirasjon fra *ArchiMate*-modellen, men har også frihet til å tilpasse struktur og visuell fremstilling for å effektivt illustrere nodenes flyt i applikasjonen.



Figur 20 Systemarkitektur (mer oversiktlig bilde i vedlegg 29)

3.5.2 Løs kodestruktur

Før utviklingen av prosjektet kom det et ønske fra Egde om at EgdeAI-Learn skulle bli pakket inn i et så kalt mono repo sammen med det eksisterende systemet EgdeAI-Chat. Et mono repo består av flere repoer/prosjekt sammen under et repo, derav mono repo. Begrunnelsen bak dette er videre utvidelse av AI i Egde. Ved utviklingen av EgdeAI-Learn så Egde dermed muligheten for å benytte mono repo.

En av hovedfordelene med å ta i bruk mono repo, er friheten til å låne kode fra andre prosjekter. EgdeAI-Chat og Learn er begge bygget i React. React er et komponent basert rammeverk, hvor hovedfunksjonen er å segmentere kode inn i komponenter som kan bli gjenbrukt i systemet. Ved utvikling av EgdeAI-Learn var det ønsket at komponenter skulle bli laget, slik at videre utvikling av AI-verktøy skal bli kjapp og smidig. Dersom Egde skulle utviklet et nytt AI-verktøy, vil det si at den nye applikasjonen kunne ha hentet komponenter fra både Learn og Chat.

Både backend og frontend komponenter er blitt segmentert og lett tilgjengelig for gjenbruk ved en senere anledning. Ved en slik tilnærming skapers en løs kodestruktur som gjør det lett å vedlikeholde flere prosjekt. Ved et eksempel hvor flere AI-verktøy hos Egde tar i bruk samme komponent, trengs det kun å gjøre endringer i selve komponenten for at alle prosjektene skal få endringen. Dette gjør mono repo i kombinasjon med React utrolig effektivt og lett å vedlikeholde, samt gjøre endringer.

3.5.3 Frontend

Ved implementeringen av frontend-designet i EgdeAI-Learn, som blir utviklet på vegne av Egde, er det av vesentlig betydning å først følge Egde sine retningslinjer. Som tidligere påpekt, representerer EgdeAI-Learn en utvidelse av Egde sine kommende AI-verktøy, som Egde ønsker å bygge på under en felles paraply. For å sikre at produktet og den underliggende koden er enkelt gjenkjennelig for ansatte og for videreutvikling, har det vært nødvendig å tilegne oss nye verktøy som Egde benytter. Sentralt i denne sammenhengen er bruken av Next.js med React rammeverket for produktet. Når det kommer til styling, har det blitt tatt i bruk Tailwind CSS kombinert med et bibliotek kalt ShadCN. Dette biblioteket inneholder ferdigbygde komponenter som ble tilpasset etter designet i Figma.

Grunnen til at Tailwind, og ShadCN er viktig, er at EgdeAI-Learn er under et mono repo sammen med EgdeAI-Chat. Ved bruk av Tailwind og ShadCN kan bruken av komponenter lett bli overført til andre prosjekter. Som nevnt tidligere, ble det ønsket av Egde at komponentene utviklet for EgdeAI-Learn skulle være med å løfte alle fremtidige prosjekt hos Egde. Gruppen utviklet altså grunnmuren for videre AI verktøy hos Egde.

For å sikre at disse komponentene kan bli brukt igjen i andre prosjekt har gruppen dokumentert alt av arbeid, kommentert i koden, og gjort koden lesbar for nye som tar over prosjektet etter oss. Biblioteket for komponentene er og ferdig designet og ligger i Egde sin Figma. Alt av komponent kode ligger implementert i Tailwind sin config fil i prosjektet. Ved å legge frem alt dette forarbeidet, vil det bli lettere å ta over prosjektet som helhet.

Gruppen fikk vite før utviklingen var startet at systemet måtte kunne bli designet ut ifra kunden sitt behov. Ved endring av komponentene i koden, det vil si, endre utseende på knapper, farge og diverse, er dette lagt opp til. Ved et godt tilrettelagt komponent bibliotek er altså systemet sammenhengende. Det at knappene er sammenhengende betyr at om en skulle utført endringer hos en knapp, vil alle knappene arve samme endring. Det betyr ikke at mulighetene for å kun endre en spesifikk knapp er borte, men muligheten for å endre alle effektivt og raskt ligger innebygd i koden. Systemet kan i den forstand bli kalt "løst" og designet så den kan tilpasses ut ifra kunden sitt behov.

Et annet eksempel på en løsning som illustrerer systemets tilpasningsdyktighet er globaliseringen i systemet. Per dags dato tilbyr EgdeAI-Learn to språk: norsk og engelsk. All tekst i systemet er kodet som nøkler, som refereres til i en globaliserings fil og oversettes deretter. Denne funksjonaliteten muliggjør integrering av et ubegrenset antall språk i systemet, forutsatt at en tilsvarende globaliserings fil med oversettelser legges til. Dette ble implementert med tanke på potensielle kunder av EgdeAI-Learn som kunne ha utenlandsk arbeidskraft, spesielt om sommeren. Ettersom EgdeAI-Learn er et opplæringsverktøy beregnet både for fast ansatte og nykommere, ble engelsk inkludert i systemet. Dette tiltaket ble gjennomført med hensyn til at det ikke nødvendigvis vil være brukere med kunnskap om norsk som benytter seg av applikasjonen.

3.5.4 Backend

Implementasjonen av backend koden i prosjektet har basert seg på Node.js rammeverket, som lar JavaScript kode kjøre i et server miljø. Dette gjelder også for Typescript, den utvidede versjonen av JavaScript som ble brukt i prosjektet. Videre har en rekke kodepakker eller kodebiblioteker blitt tatt i bruk for å dra nytte av ferdiglagde funksjoner. En god del av pakkene som har blitt tatt i bruk i prosjektet kommer fra LangChain, som er et rammeverk for Python og JavaScript applikasjoner der språkmodeller tas i bruk.

Oppstartsfasen av utviklingen i backend delen av applikasjonen gikk hovedsakelig ut på å teste ulike funksjonalitet, blant annet kommunikasjon med kunnskapsbasen. Mye av funksjonaliteten som ble testet ble klargjort for bruk i den faktiske applikasjonen. Mye av den klargjorte koden kunne senere implementeres etter hvert som både designet og frontenden av applikasjonen hadde kommet et godt stykke i gang.

Etter hvert som både designet og mye av frontenden var på plass, ble det konstant nye backend funksjoner som kunne implementeres. Mye av arbeidet gikk inn i administrator-siden, der brukere med administratortilgang kan laste opp og ned, lese av og slette innhold fra kunnskapsbasen. Funksjonaliteten relatert til opplasting og sletting av innhold kommuniserer med Azure, der det totale innholdet fra en enkelt opplasting lagres, og Pinecone, der den oppstykkede og vektoriserte dataen lagres. Den oppstykkede dataen overlapper med dataen den kommer etter, slik at det er en sammenheng mellom datastykkene. Lesingen av hva som er lagret kommuniserer kun med innholdet som er lagret på Azure.

Selve AI-en sine responser blir definert av prompt profileringen, i kombinasjon med en kontekst blokk, som består av relevant data fra Pinecone. Prompt profileringen AI-en tar utgangspunkt i, gir den ulike instruksjoner som blant annet informasjon om at den er EgdeAI-Learn, at en gruppe bachelorstudenter har utviklet den hos Egde, hovedoppgaven til selve AI-en, og litt basisinformasjon om Egde. Utenom dette blir AI-en instruert om å respondere på samme språk som den blir stilt, og at den kun skal svare på spørsmål som er relevante i forhold til informasjonen den har tilgjengelig i prompt profileringen og kontekst blokken.

Måten kontekst blokken blir satt på, er via resultater fra et søk i Pinecone dataen. Dette søket henter brukeren sin siste melding, sender gjennom en embedding-prosessen, og deretter utfører et søk gjennom Pinecone datasettet som er samtalens gjeldende kunnskapsbase. Deretter hentes ti ulike resultater ned, med hver sin poengsum mellom 0 og 1, som indikerer hvor relevant resultatet var for brukerens siste melding. Dersom et resultat hadde 0.25 i poengsum, ville dataen fra det resultatet bli tatt med i kontekstblokken. Hvis ingen av resultatene har høy nok poengsum, vil ikke AI-en ha noe datagrunnlag å svare ut ifra. Dermed ville AI-en ikke være i stand til å respondere på brukerens melding med mindre den er relevant for innholdet i forespørselens profilering, og ville i stedet gi tilbakemelding om mangelen på tilstrekkelig datagrunnlag for å kunne gi et svar.

Disse poengsummene og resultatene ble justert på i løpet av prosjektet, samt størrelsen på oppstykkningen av data. Grunnen til at dette måtte justeres på, var at AI-en ikke responderte på visse spørsmål til tross for at relevant data var til stede i Pinecone. Responsene måtte også testes en del med irrelevante spørsmål, for å forsikre at AI-en ikke svarte på spørsmål hvor relevant data manglet.

Måten mye av funksjonaliteten fungerer på er at en funksjon blir etterspurt på klient siden, sendt til server siden med den nødvendige informasjonen, og prosessert der. Etter funksjonen er gjennomført på serversiden nødvendig informasjon sendes tilbake til klient siden. Informasjonen som sendes tilbake avhenger av hva funksjonaliteten på server siden klarte. Dersom noe feilet på server siden, ble dette håndtert på en måte som ga klient siden informasjon om at noe gikk galt, og potensielt grunnen til hvorfor.

3.5.5 Implementeringsutfordringer og sikkerhetstiltak

Dette delkapitlet tar for seg implementeringsutfordringene som oppstod under opplastning på administrator-siden, altså opplastning av YouTube undertekster. Det diskuteres også i delkapitlet om sikkerhetstiltakene som ble iverksatt for lagring av totale innholdet i blob storage.

Endringer ved Youtube-opplastning

På administrator-siden kunne brukerne med administrator-tilgang laste opp undertekster fra YouTube-videoer til kunnskapsbasen. Etter implementeringen opplevde gruppen problemer da Google endret oppførselen til YouTube APIet, som førte til at funksjonaliteten til YouTube-pakken fra Langchain-biblioteket ikke lenger fungerte. Dette problemet ble hovedsakelig tilskrevet problemer med den eksterne delen av pakken, som ikke lenger kunne samhandle sømløst med de oppdaterte API-endringene.

For å imøtekomme denne utfordringen, ble to mulige løsningsstrategier vurdert. Den første innebar implementering av en midlertidig løsning, hvor utviklere ved senere anledning ville bli instruert om å gå tilbake til den opprinnelige funksjonaliteten når YouTube-pakken var blitt rettet av Langchain. Alternativt kunne YouTube-funksjonaliteten midlertidig deaktiveres inntil pakkeproblemet ble løst av utviklerne.

Beslutningen ble tatt å deaktivere YouTube-funksjonaliteten midlertidig og vente på en rettelse fra Langchain. Problemet ble ikke ordnet innen prosjektløpet av Langchain, derfor ble det revurdert å implementere en midlertidig løsning. Denne fremstillingen ble forkastet på grunn av tidsbruk og ressursprioriteringer.

Sikkerhetstiltak av lagring i blob storage

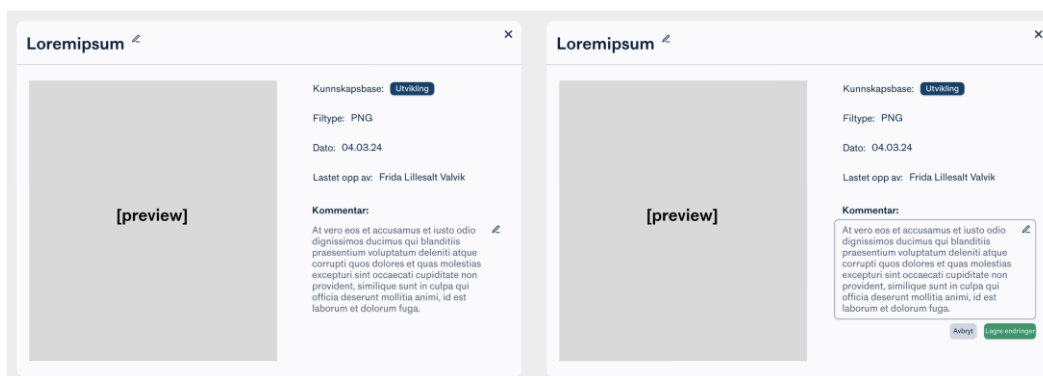
I tillegg til disse utfordringen ved YouTube-opplastning, var det nødvendig å vurdere lagringen av totale innholdet som ble lastet opp fra de ulike opplastningsmetodene opp til kunnskapsbasen. Tidligere ble dette innholdet lagret i Vercel Blob Storage.

Det ble besluttet å migrere dette innholdet til Azure Blob Storage av hensyn til sikkerhet. Årsaken til denne beslutningen var knyttet til bekymringer rundt datasikkerhet, spesielt med tanke på sensitiv og lukket informasjon som kunne bli opplastet til plattformen. Det ble observert at Vercel Blob Storage eksponerte filene for uautorisert tilgang gjennom enkel tilgang til URL-er, hvilket innebar en potensiell sikkerhetsrisiko. I kontrast til dette tilbyr Azure Blob Storage en mer lukket tilnærming, der selv om URL-er ble oppnådd, ville filene forbli utilgjengelige uten riktig autentisering. Denne endringen ble derfor ansett som et nødvendig tiltak for å ivareta datasikkerheten på plattformen.

3.5.6 Fremtidige utviklingsmuligheter

Det er designet for funksjoner som dessverre ikke har vært mulig å implementere innenfor de gitte tidsrammene for prosjektet. Det har vært ønskelig å integrere disse funksjonene, men andre prioriteringer har ført til begrensninger i utførelsen. Dette gjelder spesielt funksjonalitet for administrasjonssiden av systemet. For brukere med administratortilgang var hensikten å gi dem muligheten til å endre hvilken kunnskapsbase en fil tilhører, samt å endre filnavnet direkte i systemet.

Figur 21 viser forslag på filoversikt. Her ville brukeren kunne trykke på den blå knappen som forteller hvilken kunnskapsbase filen tilhører for å endre. Tittelen på filen ville også kunne endres ved å trykke på blyanten.



Figur 21 Forslag til fremtidige utviklingsmuligheter i filoversikt.

Det ble også diskutert mulighetene rundt handlingen å angre etter å ha slettet filer. Dette kom også frem som et ønske fra brukere under testing. Selv om implementeringen av den nevnte funksjonen ble ansett som ønskelig, ble den vurdert som noe utenfor prosjektets omfang. Dette valget ble basert på nøye vurdering av ressursbegrensninger, tidsrammer og primære målsettinger som var definert for prosjektet. Det ble i stedet konkludert med å gå for en mer forebyggende strategi. Med sikte på å forhindre at brukerne når det punkt hvor de angre på sletting av en fil, vil de heller motta tydelige og klare meldinger før slettingen gjennomføres, hvor de vil bli informert om at handlingen vil medføre permanente endringer. Brukeren vil også få en oversikt over hvilke filer de er i ferd med å slette og må godkjenne dette før filene blir slettet, for å videre begrense behovet for å måtte hente tilbake slettede filer. Som et resultat ble det bestemt at implementeringen av de nevnte funksjonene ville bli holdt utenfor det nåværende prosjektløpet, men med muligheten for å vurdere de i mulige fremtidige iterasjoner.

4. Kvalitetssikring

I dette kapittelet vil det diskuteres hvordan det har blitt forsøkt å sikre kvalitet i prosjektarbeid og produkt. Kvalitetssikringen ble utført med tanke på gruppens definisjon av kvalitet, som ble definert tidligere i delkapittel 1.4.2. Her vil det utdypes om hva som ble gjort for å sikre kvalitet i forhold til definisjonen som ble satt. Testing og evaluering av designet/bruken av applikasjonen for å sikre kvalitet vil også utdypes.

4.1 Mål for kvalitetssikring

I delkapittel 1.4.2 ble det definert tre hovedaspekter innen kvalitetssikring. Disse tre hovedaspektene, som blir forklart mer i dybde nedenfor, definerer mål for selve kvalitetssikringen som ble fulgt og forsøkt oppnådd gjennom hele prosjektløpet.

4.1.1 Produktet tilfredsstillende forventninger

Gruppen tok i bruk MoSCoW-analysen, som ble beskrevet i delkapittel 3.2.2, for å kartlegge hvilke funksjonaliteter som forventes å være til stede, samt andre funksjonaliteter som bør eller kan inkluderes. MoSCoW-metoden hjelper også å prioritere oppgaver som skal gjøres, slik at de viktigste funksjonalitetene blir implementert først. Dette verktøyet ble anvendt som et hjelpemiddel gjennom prosjektutviklingen for å sikre at det endelige produktet inneholder de forventede funksjonene, og at de viktigste funksjonalitetene ble prioritert først.

Som nevnt satt oppdragsgiver få spesifikke krav for prosjektet, men kom heller med et ønske om at gruppen utforsker og videreutvikler bruk av AI i Egde. I prosessen med å definere prosjektbeskrivelsen forsøkte gruppen å utforske og kartlegge brukerbehov og hvordan AI kunne skape verdi for brukerne. For å undersøke dette ble det holdt intervju og samtaler med potensielle brukere om bruk av AI, hvordan de tilnærmer seg ny kunnskap, og det ble også spurt om hvordan de opplevde onboarding/coach-opplegget som eksisterer på arbeidsplassen i dag. Det ble avdekket et behov for å ha enklere tilgang på informasjon, særlig for nyansatte. De brukerne som ble presentert med prosjektideen før utviklingen startet uttrykte positive tanker om en slik tjeneste og kunne se for seg at det ville gi dem verdi både som ny eller eksisterende ansatt. Gruppen har dermed utforsket og avdekket mulige bruksområder og videreutviklet AI-paraplyen i Egde. Funnene som ble gjort gjennom arbeidet ble, som nevnt i delkapittel 2.4.3, videreformidlet til Egde gjennom ukentlige stand-up-møter. Dette medførte at gruppen kontinuerlig kunne kvalitetssikre at arbeidet møtte forventningene til Egde gjennom hele arbeidsprosessen.

4.1.2 Fornøyde brukere

Gruppen har i løpet av prosjektløpet utført intervjuer, brukertester, samt eksperttesting. Testing ble utført som en iterativ prosess, med den hensikt om å veilede utviklingen i løpet av prosjektløpet. Dette tillot en kontinuerlig tilpasning av systemet basert på tilbakemeldingene fra brukerne. Ved å ha brukerne direkte involvert i utviklingsprosessen, ble det lagt vekt på å unngå antakelser om brukernes behov og preferanser. I stedet ble systemet testet og justert basert på faktiske tilbakemeldinger, med det formål om å sikre at sluttproduktet oppfylte brukernes faktiske behov og ønsker.

4.1.3 Kvalitet i kode

Ved å oppnå god kvalitet i koden er det blitt lagt opp å følge god og felles programmering skikk som har blitt lært av gruppen i tidligere emner. Følge en god struktur og arbeidsflyt med backlogg-artikler, forklaring av koden med kommentarer, kode struktur som følges likt gjennom prosjektet og testing av kode i eget miljø.

I utviklingsprosessen ble det definert en felles kodeskikk, som sikret en enhetlig og forståelig struktur gjennom hele prosjektet. Denne tilnærmingen inkluderte kommentarer i koden for å forklare funksjonaliteter, samt bruk av meningsfulle funksjonsnavn. Videre ble en standard mappe- og filnavnkonvensjon fra et av Egdes utviklingsteam, Velox, benyttet gjennom prosjektet, hvor funksjonsnavn ble formatert i pascal-case (for eksempel `function nameOfTheFunction`).

I tillegg til å følge disse konvensjonene, ble kode utviklet med hensyn til prinsippene om cohesion og coupling. Dette innebar å designe produktet slik at det var klart for raske og effektive endringer etter behov, for eksempel hvis produktet skulle tas i bruk av andre kunder. Ved å utvikle koden med fokus på prinsippene om cohesion og coupling, ble det lagt grunnlag for høy kvalitet i koden. Først og fremst bidro dette til å skape en mer modulær og robust kodebase. Ved å designe produktet slik at relaterte funksjoner var gruppert sammen og hadde lav grad av coupling, ble koden mer oversiktlig og lettere å vedlikeholde. Dette gjorde det enklere å identifisere og isolere endringer som måtte gjøres, samtidig som risikoen for utilsiktede bivirkninger eller feil ble redusert.

For å oppnå enda bedre kvalitet på koden ble det enkelte ganger praktisert pair programming. Pair programming-sesjoner ble som oftest gjennomført under sammenslåing av utviklingsgrener, hvor to/tre utviklere testet og review kode endringene på samme maskin. Pair programming, som er en agil testmetode, muliggjorde kontinuerlig skriving og testing av kode i sanntid, og bidro til å identifisere feil og forbedringsmuligheter tidlig i utviklingsprosessen. Denne praksisen styrket ikke bare kvaliteten på koden, men også samarbeidet og kunnskapsdelingen blant teammedlemmene.

4.2 Testing/evaluering av design

For å sikre kvalitet i designet har gruppen valgt å gjennomføre eksperttesting og brukertesting. Denne prosessen har vært svært viktig for å få grundig tilbakemelding på de ulike designvalgene og -løsningene i systemet.

Ekspertevaluering

Gruppen fikk mulighet til å presentere designet for en erfaren UX-designer i Egde, og dermed utføre en «ekspertevaluering». Dette møtet kom relativt tidlig i designprosessen for å sikre et godt utgangspunkt å legge videre arbeid på. Det ble presentert wireframes og et utkast av prototype for desktop og mobil, som etter møtet førte til en ny iterasjon med en ganske signifikant endring. I løpet av dette møtet ble det utført en crazy-8, hvor alle skisserte 8 mulige løsninger til et gitt problem eller tema på 8 minutter. Det valgte temaet som skulle utforskes var hvordan brukerne kan velge hvilken kunnskapsbase de ville kommunisere med. Utgangspunktet fra prototypen var at brukerne ble møtt med en landingsside når de først åpnet applikasjonen, hvor de måtte velge en kunnskapsbase før de kom inn i chatten. Hovedfunksjonen til applikasjonen er å chatte med AI-en, så det var ønskelig å unngå unødvendige steg for å komme seg dit. Ved å utføre en crazy-8 ble det utforsket mange mulige løsninger, og etter litt diskutering i plenum, kom gruppen frem til å la brukerne velge kunnskapsbase gjennom chatten. Med denne løsningen kommer brukerne rett inn i chatten og velger kunnskapsbase som første steg i samtalen.

Brukertesting

For å teste prototypene ble det utført brukertester med potensielle fremtidige brukere. Testene ble utført på kontoret med kollegaer fra ulike aldersgrupper og fagfelt. Målet med brukertestene var å få innblikk i om brukerne opplevde applikasjonen som brukervennlig, enkel å navigere, og forsto de ulike funksjonene i applikasjonen. Oppgavene som ble definert på forhånd innebar for eksempel å finne brukerguide, historikk, logge ut, og lignende for å se om de var plassert på en naturlig plass. Se vedlegg 30 for full oppgavebeskrivelse. Det ble også stilt spørsmål underveis om hva brukerne tenkte og følte. Hovedfunksjonen til applikasjonen er å snakke med AI-en, dermed var oppgavene knyttet til dette ikke bare ment for å se om de klarte å utføre en samtale. Hensikten her lå mer å få innsikt i hvordan de opplevde situasjonen; hva de synes om blant annet fargebruk, plassering av elementer og oversiktighet.

Tilbakemeldingene fra brukertestene viste at det var enkelt å forstå med liten sannsynlighet for brukerfeil og forvirringer. For brukere med administrator tilgang, er det litt flere funksjoner og muligheter i applikasjonen. Denne prototypen krevde litt mer tid til brukertester, og krevde flere iterasjoner for å få alt på plass. I tilbakemeldinger fra brukere kom det ønsker og tanker som ble nøye vurdert og førte til implementering av flere funksjoner. Gjennom brukertester ble gruppen også gjort oppmerksom på funksjoner som ikke var tatt i betraktning tidligere, men som var essensielle for brukerne. Dette inkluderer for eksempel "last ned" funksjonen, som tillater brukere å laste ned en fil som ligger i kunnskapsbasen.

Det ble holdt en ny runde brukertesting for å bestemme design av chathistorikken. Her ble det vurdert å utføre en A/B test, men etter en samtale med en designer i Egde ble det gitt anbefaling om å utføre brukertest i stedet. Denne brukertesten gikk ut på å presentere to versjoner for brukerne, versjon A og B (se vedlegg 31), og høre på tilbakemelding om hvilken de likte best og hvorfor. Resultatet fra denne runden med brukertesting viste veldig klart at majoriteten ville ha versjon B, med begrunnelsen om at det var mer oversiktlig og enklere å finne igjen tidligere samtaler.

4.3 Kvalitetssikringsaktiviteter

Designprinsipp: I tillegg til å gjennomføre brukertesting og samle feedback fra brukerne, ble det som nevnt i kapittel 3. Prosjektgjennomføring tatt i bruk designprinsipper for å sikre kvaliteten på brukergrensesnittet. Ved å ta i betraktning Benyons designprinsipper gjennom designprosessen og evaluere designet i etterkant basert på disse prinsippene, styrker gruppen sannsynligheten for gode brukeropplevelser og dermed øker kvalitet i produktet. Videre ble designet også vurdert i samsvar med retningslinjene fra WCAG, som bidro til å sikre at produktet er inkluderende og gir en behagelig opplevelse for et bredere spekter av mennesker.

Måloppnåelse: Gruppen definerte de to viktigste målene for prosjektet i prosjektbeskrivelsen (vedlegg 4). Som diskutert i delkapittel 1.4.1, var disse målene viktige for å sikre prosjektets kvalitet, men også gruppens læringsutbytte ved prosjektarbeidet.

Det ene målet var at det utførte arbeidet skulle resultere i et produkt/rammeverk som gir verdi til oppdragsgiver/prosjektkunde. Dette vil også være inkludert i vurderingen av oppnådd kvalitet i prosjektet og er knyttet til et av hovedaspektene innen definisjonen av kvalitet, nemlig at produktet tilfredsstillende forventninger (se delkapittel 1.4.2). Verdi i denne konteksten er også knyttet til utforskningen og kunnskapsbyggingen i løpet av prosjektet, ikke bare det endelige produktet. Basert på dialog med Egde ble det nevnt flere ganger at EgdeAI-Learn ble vist frem og diskutert som et reelt produkt i kundedialoger der EgdeAI var hovedfokuset. Dette beviste og verifiserte målet om at forskningen og det utviklede produktet hadde en reel nytteverdi for Egde og deres produktlinje.

Det andre målet definert i prosjektbeskrivelsen var relatert til gruppens egne læringsutbytter ved prosjektarbeidet. Dette omfattet en ambisjon om å tilegne seg og anvende nye ferdigheter og kunnskaper, samt å videreutvikle tidligere erfaring og kunnskap. Gruppen hadde for eksempel erfaring med scrum-metodikken fra tidligere prosjekter, som spilte en betydelig rolle i effektiviteten og suksessen med scrum-prosessen i dette prosjektet. Samtidig har gruppens egenskap til å bruke scrum-aktiviteten effektivt hatt en merkbar forbedring fra sprint til sprint.

Den faglige kompetansen har også hatt en merkbar effekt på gruppen. Arbeidet og forskningen med de nye tjenestene og verktøyene har styrket gruppen og enkeltpersoners kompetansenivå innenfor arbeidet med de gitte teknologidomene som ble brukt i prosjektet.

Scrum aktiviteter: Alle scrum-aktivitetene som har blitt brukt i gruppens arbeidsmetodikk har bidratt til å måle om prosjektet holder stø kurs, men gruppen mener selv at den viktigste aktiviteten for å måle kvalitet har vært weekly stand-up møtene.

Ukentlige møter med oppdragsgiver har vært en sentral faktor i kvalitetssikringen. Ved å dele fremskrittene regelmessig har gruppen vært i stand til å demonstrere kontinuerlig arbeidsflyt og fremdrift i prosjektet. Dette har bidratt til å bygge tillit og trygghet både internt i gruppen, men også mellom gruppen og oppdragsgiver, da det gir en konstant oppdatering om status i prosjektet og muligheten til å rette opp eventuelle avvik eller bekymringer tidlig i prosessen.

I tillegg har disse ukentlige møtene åpnet for hyppige interaksjoner med oppdragsgiver. Dette bidrar til muligheten for å justere kursen og gjøre nødvendige endringer i produktutviklingen mens prosessen er i gang. Dette har ført til en smidigere og mer tilpasningsdyktig tilnærming, hvor gruppen kan reagere raskt på endrede krav eller nye utfordringer underveis.

Videre har den kontinuerlige praksisen med å vise frem arbeidet som er blitt utført, ikke bare bidratt til å opprettholde et høyt nivå av motivasjon og engasjement blant gruppen, men også fremmet en følelse av tilhørighet til arbeidet som blir gjort. Samlet sett har disse møtene spilt en stor rolle for å opprettholde jevn og produktiv prosjektgjennomføring og bidratt i å sikre kvalitet i arbeidet, samt styrke samarbeid mellom gruppen og arbeidsgiver.

5. Refleksjon

Å analysere grundig hvorfor dette prosjektet utviklet seg på den måten det gjorde er en kompleks oppgave, da det involverer flere variabler og faktorer. I dette kapitlet forsøkes det å gi innsikt i de valgene som ble tatt, deres begrunnelser og generelle refleksjoner om arbeidet og læringsutbyttet.

Bakgrunn / grunnlag for suksess

Gjennom hele prosjektløpet har gruppen kunne dratt nytte av sin etablerte arbeidsdynamikk og erfaring med prosjektarbeid. Å ha en felles oppfatning om hvert gruppemedlems interessedomener og ferdigheter fra starten av prosjektet førte til at fordelingen av arbeidsoppgaver var enkel og effektiv. Prosjektet kom derfor fort i gang, og det var god progresjon fra start til slutt. Gruppens transparente forventninger og gjensidig ansvar for å opprettholde et produktivt og trivelig samarbeidsmiljø har vært avgjørende for utfallet av prosjektet.

Å velge et åpent prosjekt som kun hadde krav om å forme noe innenfor AI viste seg å være et godt valg, hovedsakelig fordi gruppen gjorde det til et godt valg. Om gruppen selv ikke hadde definert et prosjekt med nok innhold og arbeid som kunne engasjere samtlige gruppemedlemmer over lengre tid, kunne det hatt negative konsekvenser for prosjektet og bacheloroppgaven. Defineringsen av et prosjekt alle kunne bidra til kontinuerlig var essensielt for å opprettholde progresjon, motivasjon og læringsutbytte.

På bakgrunn av et slikt prosjekt var det viktig for gruppen å ha tilgang til hjelp og åpen kommunikasjon med bedriften. Å arbeide samlet og på bedriftens lokaler ga gruppen tilgang til veiledning gjennom hele prosjektforløpet, en ressurs som aktivt ble tatt i bruk.

Et sentralt aspekt ved valget av prosjekt var at resultatet skulle gi verdi for bedriften. Vi håper at verdien Egde sitter igjen med etter dette prosjektet er en enda dypere innsikt i mulighetene EgdeAI kan benytte seg av ved videre forming og utvikling.

Vi mener at Egdes valg om å gå for en åpen oppgave har ført til at vi som prosjektgruppe har fått en genuin følelse av eierskap til produktet EgdeAI-Learn. Dette samsvarer med vårt mål om å være en integrert del av prosjektet og produktet fra start til "slutt", og dermed levere noe av reell verdi til oppdragsgiveren.

5.1 Hva har gått bra? (og hvorfor)

Dokumentasjonen og beskrivelsene gjennom hele rapporten belyser i stor grad hva vi mener har fungert godt og bidratt til prosjektets suksess. Det kan derfor være utfordrende å presist identifisere hva som har fungert spesielt bra eller best med prosjektgjennomføringen. Likevel ønsker vi å fremheve noen eksempler som har fått mindre oppmerksomhet i rapporten eller som krever ytterligere refleksjon.

5.1.1 Arbeidsmetodikk

Vi har opplevd stor suksess med arbeidsmetodikken som ble tatt i bruk gjennom prosjektløpet. Ved å anvende aktiviteter som sprint planning, daily-standup og weekly stand-up, har gruppen effektivt opprettholdt struktur og fokus gjennom hele prosessen. Samtidig har denne metodikken gitt rom for utforskning og kontinuerlig tilpasninger etter behov, som vi mener var helt sentralt grunnet oppgavens åpne natur. Gjennom sprint retrospectives har gruppen muliggjort grundige refleksjoner og kontinuerlig forbedring i arbeidsprosessen.

Gjennom eksperimentering med tre ukers lange sprinter var målet for gruppen å evaluere muligheten for å opprettholde kontroll over arbeidsprosessen innenfor denne tidsrammen. Erfaringen indikerte at vi følte oss tilfreds med denne sprintlengden. Det ble oppfattet at det var mulig å oppnå betydelige fremskritt i løpet av en sprint på tre uker, samtidig som estimeringsprosessen for arbeidsoppgavene ikke var betydelig påvirket. Denne tilpasningen ble derfor betraktet som hensiktsmessig og vellykket for prosjektets fremdrift.

Som nevnt, på grunn av prosjektets åpne karakter, ville det vært svært utfordrende å forsøke å sette opp en fullstendig produktbacklogg fra start. Oppgaven for prosjektet besto av utforskning, hvor det endelige produktet ville basere seg på denne utforskningen. Det var dermed sentralt for oss å ha friheten til å justere og forme visjonen for sluttproduktet underveis i utviklingen. For å beholde retningen underveis ble det definert mål i starten av prosjektløpet, men det var usikkert nøyaktig hvor prosjektet ville ende og hvilke steg som måtte til for å nå dette endelige målet. Hadde en fullstendig produktbacklogg blitt satt helt i begynnelsen av prosjektet, ville backlogg-artiklene vært potensielt langt mer overfladiske, og muligens mer villedende for prosjektets progresjon.

Vi har opplevd at prosessen med å generere backlogg-artikler underveis i prosjektløpet som vellykket og passende for en slik åpen oppgave. Denne prosessen ble stadig bedre etter hvert som prosjektet utviklet seg, fordi vi ble flinkere til å segmentere, opprette og dokumentere backlogg-artikler underveis. Vi mener at sprint retrospective spilte en sentral rolle i vår forbedring gjennom refleksjon, tilbakemelding og finjustering. Det ble også lettere å definere nye backlogg-artikler etter hvert som vår visjon og mål ble tydeligere definert gjennom prosjektløpet.

5.1.2 Kontinuerlig dokumentering av arbeid

Gjennom hele prosjektperioden har vi gjennomført en rekke scrum-aktiviteter, møter og workshops som en del av det kontinuerlige arbeidet med å opprettholde god kommunikasjon og deling av arbeid både internt i scrum-teamet og med oppdragsgiveren. Ettersom varigheten av et slikt prosjektløp har en tendens til å strekke seg over lengre tid, har god dokumentasjon og referatføring vært avgjørende for å bevare diskusjonene, tankene og refleksjonene som har oppstått gjennom prosjektløpet.

Denne dokumentasjonen har vært spesielt viktig når det har vært behov for å gjøre rede for beslutninger som ble tatt for flere måneder siden. Eksempler på denne dokumentasjonen finner vi gjennom referat fra daglige stand-up-møter, ukesrefleksjoner, referater fra sprintplanlegging og retrospektiver, samt møterreferater fra møter med oppdragsgiveren. Disse dokumentene har vært spesielt nyttige ved utformingen av denne rapporten, da vi effektivt og raskt har hatt mulighet til å *reise tilbake i tid* og se over dokumentasjon og tanker som ble gjort for flere måneder siden. Eksempler på disse dokumentene finner man i vedlegg 5 – 8.

5.1.3 Samarbeid med Egde

I delkapittel 2.4.1 ble det beskrevet at Egde ville ta en mer passiv rolle i selve prosjektarbeidet og overlate det meste av prosjektgjennomføringen til oss. Det er likevel viktig å påpeke at Egde og ressursnettverket på kontorene har vært en uvurderlig nøkkel til prosjektets suksess. Dørstokkmilen for å be om hjelp og veiledning har vært ikke-eksisterende og har skapt et trygt miljø for læring og kompetansebygging. Gjennom Egdes flate rollestruktur har vi fått hjelp og tilbakemeldinger fra både veteraner og nye ansatte, noe som i stor grad har bidratt til vår læring og evne til å produsere gode resultater.

Gjennom de faste weekly stand-up møtene, kunne vi som nevnt tidligere, diskutere fremgangen til prosjektet og kvalitetssikre at arbeide sto til Egdes forventninger. Disse møtene var i stor grad veldig positive og fungerte som en god arena for å få tilbakemeldinger og innspill på arbeidet. Det var blant annet under et av disse møtene at anbefalingen om å migrere til Azure Blob Storage ble nevnt. Egde har virket tilsynelatende positivt overrasket over den stabile og hyppige progresjonen som prosjektet har hatt fra uke til uke. Vi tror og håper at vårt bidrag til denne åpne modellen for en arbeidsoppgave vil bli gjenbrukt hos Egde ved senere bachelorsamarbeid.

5.2 Hva kunne vi gjort bedre? (utfordringer og begrensninger)

Selv om vi er godt fornøyde med arbeidet som har blitt utført gjennom hele prosjektperioden, er det nyttig å reflektere over prosjektets flaskehals, utfordringer og forbedringspotensialer som har påvirket prosjektet i mindre eller større grad. Noen av punktene under er relatert direkte til våre prosesser og arbeidsmetoder, mens andre punkter har vært i større grad utenfor vår kontroll. Felles for alle punktene er at de har bidratt til erfaring og læring gjennom prosjektarbeidet i en eller annen form, og er derfor viktige å reflektere over i ettertid.

5.2.1 Gjennomgang av kode

Gjennomgang av endringer som er blitt utført i koden er en vesentlig viktig del som må bli gjort i et utviklingsarbeid med flere personer. I løpet av prosjektet er det blitt erfart at det ikke alltid er blitt overholdt skikkelig gjennomgang av endringer som er blitt utført. Ved feil av gjennomgang, eller dårlig utført arbeid er det mulighet for at kode går tapt i sammenslåingen av "branches", eller at katastrofale feil kan skje med applikasjonen. Dette har vært et samtaletema som har gått igjen i sprint retrospect. Grunnen til dette er at det ikke har vært strenge nok regler på gjennomgang av kode internt i scrum-teamet og i regelsettet til mono repoet.

Ting som kunne blitt gjort for å forbedre denne prosessen hadde vært "mergeblock" i GitHub. Dette er en prosess som stopper muligheten for sammenslåing av kode før en annen person enn den som ønsker endring har sett over og godkjent koden som skal inn i produksjon. Å legge til rette for en slik restriksjon har ikke vært mulig fra vår side da dette kun kan bli utført av en administrator/eier av mono repoet som vi jobbet under.

Selv om vi kontinuerlig gjennomførte forskjellige former for generell programvarestesting, inkludert enhetstesting, konsoll logging og systemtesting, kunne testing av kode blitt bedre utført i løpet av prosjektet hos Egde. Vi satte ikke opp strukturer for automatiserte testutførelser i koden. De manuelle testaktivitetene vi gjennomførte, var avgjørende for å garantere at koden som skulle inn i produksjon ikke skadet eller ødela eksisterende funksjonalitet. Før en person kunne anse et gitt arbeid som *godkjent* og klart for produksjon, måtte vedkommende gjennomføre den manuelle testprosessen både i sitt lokale kodemiljø og i en "preview deployment" opprettet av Vercel-miljøet. Selv etter all denne testingen kunne det oppstå problemer, men disse ville raskt bli oppdaget av en annen utvikler eller personen selv.

Selv om vi har utviklet med utgangspunkt i å lage forståelig kode med gode, beskrivende funksjonsnavn og variabler, er det fortsatt nyttig å dokumentere koden videre med bruk av kommentarer. Det er ingen tvil om at kommentering i koden også kunne blitt bedre. Bruk av kodekommentarer er nyttig for å skape videre forståelse i koden, både for andre utviklere og for oss selv. Ved en eventuell overlevering av koden vil det være tydeligere for nye utviklere hvordan systemet henger sammen og hva hver funksjon utgjør. Der vi har vært svake på bruk av kommentarer i koden gjennom utviklingsprosessen, har vi prøvd i løpet av de siste ukene før levering å oppdatere koden med dokumentasjon for å skape enda bedre forståelse for de som kommer til EgdeAI-Learn etter oss.

5.2.2 Tilgjengelighet fra Egde

En begrensning i prosjektløpet var den tilgjengelige tiden ansatte i Egde hadde. De var villige til å hjelpe og bidra, men vi måtte stadig tilpasse oss deres skjemaer. Det kunne gå et par dager før de hadde tid til delta i intervjuer eller bidra med kodehjelp. Den begrensede tiden til ansatte i Egde førte også til at kodehjelpsesjoner ble korte, da de hadde andre arbeidsoppgaver å prioritere.

For å takle denne begrensningen, ble gruppen ekstra effektive på å utnytte tiden ansatte i Egde hadde til rådighet. Det innebar å planlegge og gjennomføre litt forhåndsarbeid slik at gruppen var klar med spørsmålene eller problemstillingen før møte med dem. Kodehjelpsesjoner ble unyttet effektivt ved å sende kodekontekst på forhånd, som var funksjonalitetsproblemer som krevde assistanse. Denne tilnærmingen førte til at ansatte i Egde kunne forberede seg, slik at de deretter kunne bidra med målrettet hjelp.

Vi ønsker å uttrykke at vi er takknemlige for den tiden Egde ansatte kunne avse. Deres innsats og bidrag, selv om det var begrenset, viste seg å være av stor hjelp og nytte for fremdriften av prosjektet. På lang sikt, selv om det var utfordrende å håndtere denne begrensningen, lærte vi verdifulle leksjoner om effektiv tidsstyring, fleksibilitet og samarbeid på tvers av tidsbegrensninger.

5.2.3 Estimering og faktisk tidsbruk

I arbeidet med å estimere tidsbruken på ulike oppgaver, har gruppen erfart både over- og underestimering av arbeidstiden. Som nevnt tidligere i delkapittel 2.6, ble det valgt å bevisst overestimere arbeidstid. Gruppen ble imidlertid bedre på å estimere utover prosjektløpet på grunn av økt kunnskap, og dermed en dypere forståelse av hvor lang tid arbeidsoppgavene faktisk ville ta. I tilfeller hvor arbeidsoppgaver har blitt fullført betydelig raskere enn forventet, kan dette tilskrives en feilvurdering i estimeringen vår, og ikke på grunn av den bevisste overestimeringen.

På den andre siden har det også vært tilfeller der gruppen har underestimert tidsbruken, som for eksempel under implementeringen av Azure Blob Storage. Den faktiske arbeidstiden viste seg å være betydelig høyere enn estimert. Dette fremstilte at den bevisste overestimeringen ikke førte til at arbeidstid aldri ble underestimert. Det bevisste overestimatet gikk som sagt gjerne bare en time eller to over det gjennomsnittlige estimatet, så hvis en arbeidsoppgave tok mye mer tid enn forventet, så ville den forventede arbeidstiden vært underestimert.

Samlet sett har disse erfaringene lært gruppen verdifulle leksjoner om viktigheten av å balansere realistiske estimater med mulige utfordringer som kan oppstå underveis. Det har vært utfordrende å estimere arbeidsoppgaver man ikke har kjentskap til. I de fleste tilfellene av feilestimering var det grunnet overestimering, men det var også noen tilfeller av underestimering, spesielt relatert til Azure Blob Storage migrasjonen. Begge typer feilestimering kan være problematiske, men vi opplevde det som mer gunstig å heller overestimere ved planleggingen av arbeidet. Dette ville gi oss en større sjanse til å ha mer tid til overs enn for lite.

5.3 Hva har vi lært & hva ville vi gjort annerledes?

I dette delkapittelet blir det tatt opp temaer som vi har lært fra dette prosjektet. Det blir også tatt opp aspekter ved prosjektet som vi ville gjennomført annerledes, dersom vi skulle gjennomført en ny iterasjon av det samme prosjektet.

5.3.1 Hva har vi lært?

Som en følge av prosjektet, har gruppen virkelig fått sett verdien i å følge en godt definert arbeidsplan og metodikk. Selv i et prosjekt som dette, som kan anses som åpent og eksplorativt, har det vært essensielt å holde god styring for å sikre god progresjon og kvalitet. Selv om arbeidsmetodikken som ble valgt var langt mer agil enn alminnelig scrum, var det fortsatt på plass gode styringsmekanismer for å sikre at vi gikk riktig vei.

Gruppens kommunikasjonsevne har også utviklet seg, og blitt enda sterkere enn den var ved starten av prosjektet. Aktiviteter som daily stand up og sprint retrospective ble bare bedre og enklere å gjennomføre utover i prosjektet. Den generelle forståelsen rundt prosjektarbeid har økt for samtlige gruppe-medlemmer, og kan tas med videre inn i fremtidige prosjekter.

Utenom utvidede ferdigheter og erfaringen innen prosjektarbeid, har gruppen oppnådd god forståelse for utvikling av LLM-basert AI tjenester og arkitekturen rundt disse. Vi hadde en arkitektur å gå ut ifra, inspirert av EgdeAI-Chat, men mye måtte fortsatt utforskes og bli løst på egenhånd. Det er selvfølgelig noen som har jobbet mer direkte mer med selve AI-delen av applikasjonen enn andre, men alle gruppe-medlemmene har oppnådd en god forståelse hvordan hoveddelene av applikasjonen henger sammen.

5.3.2 Hva ville vi gjort annerledes?

Gjennom refleksjonen på vårt prosjekt og dets utfall, har vi identifisert flere områder der vi ser muligheten til å gjøre justeringer for å optimalisere resultatet.

Workshop-møter

I tidligere prosjektarbeid har vi erfart stor verdi av å dele nyoppnådd kunnskap og kompetanse på tvers av faggruppene i scrum-teamet gjennom faste workshop-møter. Dette var en praksis vi ønsket å videreføre da vi visste at vi ville generere mye ny kunnskap gjennom prosjektarbeidet. I løpet av prosjektperioden gjennomførte vi to slike workshop-møter denne våren. Selv om antallet var lavere enn forventet, ga disse møtene likevel stor verdi for arbeidet som ble utført på tvers av faggruppene. Grunnen til at det kun ble arrangert to slike møter, var at vi ikke var flinke nok til å prioritere deres verdi inn i sprintplanleggingen. Ved en ny anledning hadde vi fast arrangert flere workshops internt for gruppen.

Brukertesting

Det ble anvendt en iterativ tilnærming til brukertesting gjennom prosjektløpet, med flere tester gjennom ulike stadier for å sikre utviklingen av et brukervennlig grensesnitt. I starten ble brukertester gjennomført for å evaluere hvordan brukerne opplevde systemer, og ved betydelige endringer eller implementasjon av nye funksjonaliteter ble ytterlige tester gjennomført. Det var ønskelig å utføre en runde med brukertester helt mot slutten av prosjektløpet for å evaluere det endelige produktet. Disse testene skulle ideelt sett gjennomføres med personer som hadde begrenset kjennskap til systemet på forhånd. Dessverre ble gjennomføringen av disse testene ikke mulig grunnet andre prioriterte oppgaver.

UX-modeller

Vi kunne ha forbedret oss ved å integrere empatikart, personas og andre UX-modeller mer konsekvent gjennom hele utviklingsprosessen. Selv om vi brukte disse verktøyene aktivt i tidligere faser i prosjektløpet, er det viktig å holde disse modellene i bakhodet under utviklingen for å sikre at vi fortsatt adresserer brukernes behov. Mer fokus på disse ville ha bidratt til å bedre opprettholde brukerfokus gjennom hele prosessen og sikre at vår endelige løsninger reflekterte brukernes behov og ønsker.

React komponenter

Grunnet nye verktøy og bratt læringskurve ble mye kunnskap rundt React og Tailwind utvikling lært utover sprintene. I starten av prosjektet ble styling av komponenter gjort på en måte hvor en og en komponent ble lagt inn i systemet ved behov. Det som ville blitt gjort annerledes nå er at både UX designer og FrontEnd utvikler skal sammen fra start, lage en oversikt over alt av fargekode navn, variant navn på komponenter og diverse både i Figma og i systemet for å få en skikkelig grunnmur på prosjektet før utviklingen.

Komponentene i applikasjonen burde også ha blitt utviklet tidligere i prosjektet ved å segmentere koden fra begynnelsen av. Dette hadde ført til at vi kunne dratt nytte av hovedpoenget med React, som er at man skal kunne gjenbruke kode i systemet via komponenter. Grunnen til at dette ikke skjedde i starten, var mangelen på kunnskap rundt bruken av React.

Utvikling av kode

Rundt den generelle utviklingen og koding av systemet var det en del vi kunne og burde ha gjort annerledes. Koden ble til en viss grad skrevet etter en standard, der navn på filer, funksjoner og lignede ble definert. Dette, i kombinasjon med beskrivende navngivning, ble fulgt i mesteparten av koden, men det var tilfeller der selve formateringen på koden måtte ryddes opp i til slutt.

Kommentarer for å beskrive kodens hensikt var noe som ble nevnt gjentatte ganger i sprint retrospective, men vi var ikke gode nok på å skrive kommentarer i koden fra begynnelsen av. Det var hovedsakelig i den siste sprinten at de fleste kommentarene i koden ble skrevet, og førte til at kvaliteten på kommentarene kanskje ikke ble like bra som hvis kommentarer hadde blitt skrevet med en gang. Dette, i tillegg til tankene våre rundt gjennomgang av kode (se delkapittel 5.2.1) ville ført til bedre kvalitet i koden og utviklingsprosessen.

Mye av koden relatert til funksjonalitet ble skrevet fra bunnen av. Noe kunne nok blitt hentet direkte fra EgdeAI-Chat applikasjonen gjennom mono repo strukturen. Tidlig i prosjektløpet ble det klart at Chat hadde en ganske annerledes struktur og tilnærming, som førte til tanken om at koden måtte skrives fra bunnen av. Det var ikke nødvendigvis alltid tilfellet, spesielt for tilkoblinger til Pinecone og OpenAI. Hadde prosjektet blitt ført på nytt, hadde vi nok satt oss litt mer inn i strukturen til Chat, og prøvd å gjenbruke mer kode som var relevant for Learn.

Azure i stedet for Vercel

En sentral refleksjon angående prosjektet er migrasjonen til Azure Blob Storage for lagring av blob-data. I stedet for å velge Vercel Blob Storage til å begynne med, ville det vært bedre om Azure hadde blitt tatt i bruk med en gang. For at dette skulle skjedd, måtte vi ha gjort litt mer undersøkelse rundt sikkerhet til Vercel og heller sett på andre alternativer, som Azure. Beslutningen om å velge Vercel førte til en forsinkelse i utviklingen, grunnet den nødvendige migrasjonen til Azure. Dette resulterte i tap av tid og muligheten til å identifisere og håndtere utfordringene knyttet til Azure-plattformen tidligere i prosessen.

Hadde vi valgt Azure til å begynne med, ville vi ha blitt konfrontert med utfordringene ved implementasjon på et tidligere stadium. Dette ville ha gitt oss muligheten til å adressere disse utfordringene raskt og effektivt, og dermed kunne vi ha minimert forsinkelsene og komplikasjonene som kom senere i prosjektet på grunn av Azure implementasjonen.

Aktiv bruk av MoSCoW

En siste endring vi ville ha gjennomført er en mer omfattende bruk av MoSCoW-modellen i vårt arbeid. I et prosjekt med såpass åpen karakter og variasjon i krav, burde vi ha utnyttet MoSCoW-modellen som et verktøy for å prioritere krav. Det ville vært hensiktsmessig å oppdatere og justere MoSCoW-prioriteringene kontinuerlig i tråd med endringer i prosjektets omfang og mål. Dette ville ha gitt oss muligheten til å tilpasse oss mer ved nye krav eller utfordringer som oppstod underveis.

6. Avsluttende ord

Med det er fem måneder med prosjektarbeid over, og ikke minst tre år med bachelorutdanning. Gjennom rapporten har vi forsøkt å belyse prosjektarbeidet som ble gjennomført i samarbeid med Egde Consulting våren 2024. Det er utvilsomt mye vi kunne ha diskutert i dette arbeidet, men vi har forsøkt å destillere ut de viktigste punktene for å gi en god forståelse av prosjektet, applikasjonen og arbeidsmetodikken som ble benyttet i arbeidet. I vedlegg vil man finne utvidet informasjon og detaljer om de ulike delene av prosjektet som det ikke var plass til i rapporten.

Vi har uten tvil lært utrolig mye nytt, og er enormt stolte av sluttresultatet for EgdeAI-Learn. Kunnskapen og kompetansen vi har utviklet gjennom dette prosjektet vil være utrolig verdifull i vårt videre arbeid. Å få muligheten til å utforske mulighetene innen et så dagsaktuelt og spennende fagfelt som kunstig intelligens har gitt oss verdifull innsikt i fremtidige løsninger og utvikling i IT-landskapet. For vår del er det ingen tvil om at denne oppgaven og samarbeidet med Egde var det riktige valget for oss, og vi håper vårt arbeid vil bidra til videre verdiskapning hos Egde.

Igjen, vi er utrolig takknemlige ovenfor alle som har bidratt til arbeidet vårt gjennom disse månedene.

Takk for oss,

Gruppe 9

7. Litteraturliste

Bruton, L. (2022). What are UX personas and what are they used for? UX Design Institute.

<https://www.uxdesigninstitute.com/blog/what-are-ux-personas/>

Coursera. (2023). What is a User Story? <https://www.coursera.org/articles/what-is-user-story>

Dingsøy, T. (2024). Scrum (informatikk). SNL. <https://snl.no/scrum - informatikk>

Engesæt, J., Heidari, H., Valvik, F. L., Morka, H. C. & Tangeraas, I. M. (2023). IS-310 Prosjektgjennomføring - Prosjektbeskrivelse. Upublisert innlevering.

Figma. (u.å.). Empathy map: A guide to user attitudes and behaviors.

<https://www.uxdesigninstitute.com/blog/what-are-ux-personas/>

Figma. (u.å.). What is prototyping. <https://www.figma.com/resource-library/what-is-prototyping/>

Figma. (u.å.). What is wireframing? <https://www.figma.com/resource-library/what-is-wireframing/>

Gundersen, D., & Halbo, L. (2018). Kvalitet. Store Norske Leksikon. <https://snl.no/kvalitet>

Henry, S. L. (2024). WCAG 2 Overview. <https://www.w3.org/WAI/standards-guidelines/wcag/>

IBM. (n.d.). What Is a Vector Database? <https://www.ibm.com/topics/vector-database>

Intellectsoft. (2023). Guide to System Development Life Cycle.

<https://www.intellectsoft.net/blog/what-is-system-development-life-cycle/>

Langchain. (n.d.). Introduction. Retrieved April 30, 2024, from

https://js.langchain.com/docs/get_started/introduction

LeanIX. (u.å.). What is ArchiMate? [https://www.leanix.net/en/wiki/ea/what-is-](https://www.leanix.net/en/wiki/ea/what-is-archimate#:~:text=As%20organizations%20strive%20to%20align,effective%20communicatio)

[n%20and%20decision%2Dmaking](https://www.leanix.net/en/wiki/ea/what-is-archimate#:~:text=As%20organizations%20strive%20to%20align,effective%20communicatio)

Mathiassen, L., Munk-Madsen, A., Nielsen, P. A. & Stage, J. (2018). Object oriented analysis and design (2. edition). Metodica ApS.

MDN Web Docs. (2023, December 2). Blob. Retrieved April 30, 2024, from

<https://developer.mozilla.org/en-US/docs/Web/API/Blob>

Microsoft Support. (u.å.). Prosjekttrekanten. Microsoft. [https://support.microsoft.com/nb-](https://support.microsoft.com/nb-no/topic/prosjekttrekanten-8c892e06-d761-4d40-8e1f-17b33fdcf810#articleFooterSupportBridge=communityBridge)

[no/topic/prosjekttrekanten-8c892e06-d761-4d40-8e1f-17b33fdcf810#articleFooterSupportBridge=communityBridge](https://support.microsoft.com/nb-no/topic/prosjekttrekanten-8c892e06-d761-4d40-8e1f-17b33fdcf810#articleFooterSupportBridge=communityBridge)

NVIDIA. (n.d.). What are Large Language Models? Retrieved April 30, 2024, from

<https://www.nvidia.com/en-us/glossary/large-language-models/>

Rolstadås, A. (2020). Prosjektplanlegging. SNL. <https://snl.no/prosjektplanlegging>

Rojas, J. (2023). Learn How to Use Sketching as an Ideation Method. [https://www.interaction-](https://www.interaction-design.org/literature/article/etch-a-sketch-how-to-use-sketching-in-user-experience-design)

[design.org/literature/article/etch-a-sketch-how-to-use-sketching-in-user-experience-design](https://www.interaction-design.org/literature/article/etch-a-sketch-how-to-use-sketching-in-user-experience-design)

- Schwaber-Cohen, R. (n.d.). What is a Vector Database & How Does it Work? Use Cases + Examples. Pinecone. <https://www.pinecone.io/learn/vector-database/>
- Scrum.org. (u.å.). Introduction the the Scrum Events. <https://www.scrum.org/resources/introduction-scrum-events>
- Scrum.org. (u.å.). Learn About the Scrum Sprint Event. <https://www.scrum.org/resources/what-is-a-sprint-in-scrum>
- Scrum.org. (u.å.). What is Scrum. <https://www.scrum.org/learning-series/what-is-scrum/the-scrum-team>
- Scrum.org. (u.å.). What is a Product backlog. <https://www.scrum.org/resources/what-is-a-product-backlog>
- Scrum.org. (u.å.). What is a Product Owner. <https://www.scrum.org/learning-series/product-owner/what-is-a-product-owner->
- Scrum.org. (u.å.). What is a Scrum Master. <https://www.scrum.org/learning-series/scrum-master/>
- Scrum.org. (u.å.) What is a Sprint Backlog. <https://www.scrum.org/resources/what-is-a-sprint-backlog>
- Uxcel. (u.å.). Affordance. <https://app.uxcel.com/glossary/affordance>
- UX Glossary. (u.å.). Consistency. <https://www.uxglossary.com/terms/consistency/>
- VanZandt, P. (2023). What is a MoSCoW Analysis? Definition, Use Guide, and Analysis. Ideascale. <https://ideascale.com/blog/moscow-analysis-definition/>

8. Vedlegg

Vedlegg 1 - Uttalelse fra oppdragsgiver



Bacheloroppgave UiA, våren 2024

Vi i Egde har hatt gleden av å samarbeide med en eksepsjonell gruppe bachelorstudenter fra Universitetet i Agder gjennom vårsemesteret 2024. Gruppen ble tildelt en utfordrende og åpen oppgave under vårt "AI i Egde"-initiativ, med mål om å utforske og utvikle innovative AI-baserte verktøy og tjenester. Oppgaven, som ble presentert under RefreshIT i oktober 2023, var å "gjøre noe gøy med AI". Dette ledet til utviklingen av EgdeAI-Learn, et prosjekt som har overgått våre forventninger på alle måter.

Vi ønsket oss en gruppe som fikk et "ekstremt eierskap" til oppgaven, og derfor måtte definere mye av oppgaven selv. Vi fikk det vi så etter, og kommer til å følge samme suksess-formel for fremtidige bachelorgrupper.

Bachelorgruppen har vist en imponerende evne til å ta initiativ og jobbe selvstendig, samtidig som de har vært flinke til å involvere oss i prosessen. Deres engasjement og innovative tilnærming har resultert i praktiske løsninger som allerede er tatt i bruk i våre AI-produkter til stor suksess hos våre kunder.

De har ikke bare utmerket seg i programmering og teknisk dyktighet, men også i prosjektstyring, inkludert utforming av brukerhistorier og effektiv oppgavefordeling. Samarbeidet har vært en lærerik opplevelse for oss i Egde, og vi har sett stor verdi i de nye perspektivene og ideene de har brakt til bordet.

Gruppens dedikasjon og hardt arbeid har resultert i et sluttprodukt som vi er stolte av å ha vært en del av. Det har vært en fryd å følge deres progresjon og se deres løsninger komme til liv.

Vi i Egde er strålende fornøyd med resultatet og ser frem til å se hva fremtiden bringer for disse talentfulle individene. Vi anbefaler dem på det varmeste for fremtidige muligheter, og er sikre på at de vil fortsette å gjøre en fremragende innsats i sine kommende utfordringer.

Med vennlig hilsen

Øyvind Brekkhus Sandåker
Teknologileder i Smia, Egde Consulting AS

Egde Consulting AS
Terje Levås vei 1, 4879 Grimstad
Tlf. +47 37 80 00 00
Org. nr. 917 051 859 MVA
www.egde.no

Kontorer: Terje Levås vei 1, 4879 Grimstad | Gravåne 16, 4610 Kristiansand | Dokkiveien 11, 3920 Porsgrunn

Vedlegg 2 – Egenvurdering

Engesæt, Jostein: Ansvarlig for å lede frontend-utviklingen i prosjektet i land. Jobbet mye med å bli godt kjent med rammeverket React. Jobbet med segmentering av komponenter og ferdigstilte dem basert på prototypdesign i Figma. Bistått ved brukertesting og brukerintervjuer som skribent og lytter. I tillegg har jeg bistått som en støttespiller i UX-designet, bidratt med innspill om gjennomførbare av ulike løsninger samt kommet med egne tanker og meninger rundt implementering av nye funksjoner. Hovedsakelig jobbet med koding av frontend, feilretting og diverse oppgaver helt frem til det ble innført code freeze. Etter dette har jeg deltatt i skrivingen av bachelorrapporten. For å nevne arbeid jeg har gjort: Implementert landingssiden, utviklet frontend av chat funksjonen på landingssiden. Fått på plass historikk og klargjort den for backend. Laget adminsiden med modaler for opplasting av filer og ny kunnskapsbase. Laget tabellen på adminsiden og gjort den klar for å lese data. Utviklet applikasjonen responsivt for både data, nettbrett og mobil. Laget EgdeAI-Learn sin logo. I tillegg til dette har jeg jobbet med å legge på nødvendige ting som feedback til brukeren, sperre, og universell utformet systemet, så det kan bli brukt av mangfoldet. Dette inkluderer aria labeling, globalisering, lagt opp for å kunne bruke tastatur, skjermleser, og verktøytips i systemet.

Heidari, Hassan: Jeg har jobbet hovedsakelig med backend utvikling gjennom prosjektløpet. Mitt arbeidsområde i backed har vært å implementere funksjonalitet knyttet til administrator siden. Dette innebærer å utvikle funksjonalitet for å opplaste kunnskapsdata til AI via PDF eller web URL, samt jobbet med lagringshåndtering for opplastede data. Jobbet med å utvikle løsninger for å lagre opplastede data i vektorisert format i Pinecone, samtidig lagre filene i helformat i Azure Blob. Dette involverte arbeid med embedding og chunk slicing. I tillegg til å jobbe med opplastningsfunksjonaliteten, har jeg også fokusert på å styrke administrasjonen av kunnskapsbasen ved å implementere funksjoner for å liste opp de opplastede filene. Dette gir brukeren en oversikt over tilgjengelige filer og dokumenter. Har også arbeidet funksjonaliteten til å opprette kunnskapsbaser. Utover det har jeg jobbet med å dokumentere prosjektløpet i rapporten.

Morka, Hans Christian: Jobbet hovedsakelig med utviklingen av backend delen av applikasjonen. Arbeidet mitt har vært å få på plass funksjonalitet relatert Azure Blob Storage, Pinecone og selve AI-en. På administratorsiden jobbet jeg med opplasting av data, forhåndsvisning av filer/URLer, nedlastning av filer, sletting av filer, kommentering til filer, PDF-generering for beskrivelse, setting av brukernavn til opplastede filer og tilbakemeldinger til brukeren. På chat/hovedsiden implementerte jeg funksjonalitet for chat-historikk og prompt-wizard/forslag. I slutten av kodedelen av prosjektet gikk jeg gjennom koden og fikset formatering og la til kommentarer. Jeg skrev litt på rapporten i begynnelsen av prosjektet, men jeg fortsatte ikke å skrive igjen før vi hadde sagt oss ferdige med kodingen.

Tangeraas, Ivar Michael: har hatt ansvaret som kombinert prosjektleder og Scrum-master gjennom bachelorprosjektet. I prosjektlederrollen fulgte han opp backloggen gjennom prosjektløpet og opprettholdt kontakten mellom scrum-teamet og oppdragsgiveren. Som Scrum-master arrangerte,

ledet og dokumenterte han scrum-aktiviteter gjennom prosjektløpet, og skrev også ukentlige refleksjoner på slutten av hver uke.

Som utvikler jobbet Michael innen fullstack, og han tok på seg oppgaver både innen backend og frontend. Med grunnleggende kompetanse og erfaring innen implementeringen av AI-chat-systemer, bidro han naturlig til backend-arbeidet, men han ville også utfordre seg selv med å jobbe mer med frontend også.

I utviklingsarbeidet implementerte han følgende elementer:

- Autorisering og innlogging ved hjelp av Clerk, inkludert administrasjon av administratorbrukere.
- Opprettelse og implementering av brukerguide-sider.
- Implementering av onboarding-guiden.
- Implementering og vedlikehold av prompt profiling.
- Implementering av markdown visning og code-blocks i chat-response
- Implementerte og designet error-handling sider 404 og 500

I bachelorrapporten har Michael skrevet under flere delkapitler, men har dedikert mye tid spesielt til kapittel 2.

Valvik, Frida Lillesalt: I løpet av dette prosjektet har jeg hatt rollen som UX-designer og ledet arbeidet med å forstå og imøtekomme brukernes behov. Mine hovedoppgaver har vært blant annet å samle inn informasjon om målgruppen gjennom grundige undersøkelser og analyser, inkludert utvikling av personas, brukerhistorier og empatikart. Basert på denne innsikten har jeg ledet utviklingen av designløsninger, inkludert skisser, wireframes og prototype. Jeg har aktivt anvendt prinsippene for brukersentrert design og tilgjengelighet for å sikre at produktet ikke bare er estetisk tiltalende, men også funksjonelt og brukervennlig for alle brukere. Jeg har også gjennomført flere runder med brukertesting for å validere og iterere på designvalgene. I tillegg har jeg bidratt til dokumentasjon av prosjektløpet i rapporten.

Vedlegg 3 - Gruppekontrakt

Gruppekontrakt

Vi, gruppe 9, består av følgende medlemmer:

Engesæt, Jostein	josteinen@uia.no
Heidari, Hassan	hassanh@uia.no
Morka, Hans Christian	hanscm@uia.no
Tangeraas, Ivar Michael	ivarmt@uia.no
Valvik, Frida Lillesalt	fridalv@uia.no

Sammen skal gruppe 9 jobbe om følgende prosjekt:

Bacheloroppgave hos Egde: EgdeAI-Learn

Bakgrunn og motivasjon

Gruppe 9 er sammensatt av medlemmer som har vært aktivt engasjert i et pågående samarbeid gjennom flere semestre innenfor bachelorprogrammet i IT og Informasjonssystemer. Denne kontinuerlige samarbeidsprosessen har vært fundamental for å utvikle en kollektiv kompetansebase og et solid fundament for gruppearbeid.

Hvert gruppemedlem har gjennom studiet opparbeidet seg betydelig faglig kompetanse som blir forsterket gjennom samarbeidet i gruppen. Det har blitt etablert en arbeidsdynamikk som effektivt tar i bruk hver enkelt medlems unike evner, erfaringer og styrker. Gruppen legger vekt på et inkluderende og effektivt arbeidsmiljø, som danner grunnlaget for vellykket gjennomføring av prosjekter og oppgaver.

Gruppen er bestemt på å videreføre denne vellykkede samarbeidsmodellen til vår bacheloroppgave, hvor vi forplikter oss til å levere et arbeid av høy kvalitet som reflekterer vårt kollektive engasjement og vår samlede kunnskap innen IT og Informasjonssystemer.

Bestemmelser om fremmøte og samarbeid

Jeg forplikter meg ovenfor gruppen min, til at jeg vil:

- Følge forelesninger, og delta i gruppearbeid, samlinger og øvelser, be om/ta imot veiledning.
- Respektere kravet om minst 80% fremmøte. Gi beskjed til gruppeleder i forkant av avtalt møtetidspunkt hvis man ikke kan møte.
- Overholde frister og levere oppgaver som publiseres på Canvas, innen fristene.
- Sørge for at arbeid fordeles jevnt mellom gruppemedlemmer, men samtidig utnytte hver enkelt students spesielle ferdigheter og bakgrunn, samtidig som å styrke manglende kunnskap hvor det er nødvendig.
- Vi holder aktiv kommunikasjon med hverandre dersom noen står fast og behøver hjelp.
- Vi holder oss oppdaterte på informasjonsflyten på Discord, Teams, backlogg, og e-post ved å sjekke innom jevnlig.
- Vi tar selv ansvar og initiativ til og for egen læring utenfor gruppetimene.
- Vi forplikter oss til å jobbe opp mot å produsere resultater som tilfredsstillende måloppnåelse av høyeste grad innen emnets læringsmål og karakter.

Evt. konflikter rundt samarbeid og innsats søkes løst gjennom diskusjon i gruppen. Fører ikke dette frem, kontaktes lærerne eller studierådgiver, så snart som mulig og før deadlines.

Andre bestemmelser

Arbeidstider vil normalt ligge på 7 timer (inkl. pauser) mandag til fredag fra klokken 09:00 til 16:00. Onsdager settes i utgangspunktet av til rapportskrivning og UiA-relaterte temaer, med valgfrihet mellom å møte på kontoret eller jobbe fra hjemmekontor. Utenom onsdager tillates også hjemmekontor, men det oppfordres til å møte fysisk på kontoret. Uavhengig av hvor man jobber skal hvert gruppemedlem delta på daily standup klokken 09:05.

Gruppeleder leder gruppemøtene og sørger for at alle er med i arbeidet. Er det noe som må diskuteres, skal gruppeleder fungere som ordstyrer, og sørger for at alle blir hørt. Etter en diskusjon av en lengde som passer til temaets viktighet, avveid mot behovet for å ta en beslutning og komme videre i prosjektet, tas det en omforent beslutning. Er det uenighet, foretas det en avstemming. Etter at beslutninger er fattet skal alle forholde seg lojalt til disse.

Da vi følger Scrum arbeidsmetodikken, vil også gruppeleder fungere som Scrum Master og holde styr over generelle scrum arrangementer underveis i prosjektløpet. Gruppeleder er gruppens kontaktperson utad, og hovedansvarlig for fremdrift og for at frister for innlevering blir overholdt. Gruppelederen skal også holde kommunikasjonskanaler oppdatert med viktig informasjon fra uke til uke.

Rollefordeling

Jostein	Front-End
Hassan	Back-End
Hans Christian	Back-End
Michael	Prosjektleder / Full-stack
Frida	UX-Designer

Sted, dato:

Kristiansand, 08/01/24

Underskrift:

Hans Christian Mørka Plassan Fleidri

Frida Lillesøtt Kalik M Tangeraaas

Jostein Engeset

Vedlegg 4 - Prosjektbeskrivelse



Gruppenummer: Gruppe 9

Navn på gruppemedlemmer:

Jostein Engesæt: josteinengesaset@gmail.com / josteinen@uia.no

Hans Christian Morka: hcalt12345@gmail.com / hanscm@uia.no

Ivar Michael Tangeraas: michael.tangeraas@gmail.com / ivarmt@uia.no

Frida Valvik: frida.valvik@icloud.com / fridalv@uia.no

Hassan Heidari: has.hei2002@gmail.com / hassanh@student.uia.no

FORSIDE

ved besvarelse av hjemmeeksamen, semesteroppgave, rapport, essay m.m.

Emnekode:	UIA IS-310
Emnenavn:	<i>IS-310-1 23H Prosjektgjennomføring</i>
Emneansvarlig (normalt faglærer):	<i>Hallgeir Nilsen</i>
Innleveringsfrist/ tidspunkt:	<i>28. november 2023</i>
Antall ark inkl. denne forside:	<i>8 sider + forside, litteraturliste og vedlegg. Totalt 9 sider.</i>
Merknader/Tittel på oppgave:	<i>Prosjektbeskrivelse</i>

Jeg/vi bekrefter at jeg/vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	Ja✓	Nei
Kan besvarelsen brukes til undervisningsformål?	Ja✓	Nei
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	Ja✓	Nei



IS-310 - Prosjektgjennomføring

Prosjektbeskrivelse

Produsert av Gruppe 9 – 28.11.23

Forord

Vi var tidlig innstilt på at faget IS-310 - Prosjektgjennomføring, ville spille en avgjørende rolle for det kommende vårsemesteret og det endelige bachelorarbeidet. Dette kommende arbeidet, som skal gjennomføres våren 2024, markerer avslutningen på vårt nåværende akademiske studieopphold ved Universitet i Agder (UiA) for mange av oss. Det har potensial til å fungere som et direkte springbrett ut i arbeidslivet, hvor vi kan anvende de kunnskapene og ferdighetene vi har tilegnet oss gjennom vårt engasjement i løpet av seks bachelorsemaster ved Universitet i Agder.

Derfor var det desto viktigere for oss i Gruppe 9 å finne samarbeidsmuligheter med en seriøs, dyktig og interessant bedrift, der vi sammen kunne utarbeide et godt profesjonelt og akademisk resultat som avslutning på vår bachelorgrad.

I kjølvannet av nok et spennende høstsemester ved UiA, med aktiv gruppe promotering, videoinnspillinger, RefreshIT deltakelse, samtaler, intervjuer, samt utallige utsendelser og oppfølging av e-poster, kan medlemmene i gruppe 9 endelig puste ut og betrakte vår innsats i faget IS-310 med et tilfreds smil og god samvittighet.

Bachelorprosjekt – AI hos Egde

Etter flere samtaler og intervjurunder med spennende IT-bedrifter i Agder, sto vi ovenfor valgets kval om hvilket prosjekt og bedrift som passet oss i Gruppe 9 best. Etter mye drøfting og solid magesfølelse tok vi beslutningen om å inngå et samarbeid med Egde Consulting (som videre vil bli referert til som kun Egde) for vårt bachelorprosjekt, våren 2024.

Kort oppsummert er Egde et nyere etablert konsulenthus fra 2016, som tilbyr dyktig erfaring innen alt fra rådgivning, prosjektledelse, utvikling, UX og mye mer. Med omtrent 100 ansatte fordelt på kontorer i Kristiansand, Grimstad, Porsgrunn og Tromsø, jobber Egde aktivt med å *“forene mennesker og teknologi”* (Egde, 2023).

I motsetning til mange av de andre bedriftene som deltok på RefreshIT-konferansen i høst, hadde ikke Egde en klar idé eller prosjektpitch for hva de ønsket å samarbeide om, annet enn at de ønsket å utforske bruken av kunstig intelligens (KI, eller AI på engelsk). I samtaler med vår kontaktperson hos Egde kom det frem at de ønsket å etablere og videreutvikle bruken av AI i deres tjenester og portefølje. Eller som det ble formulert i møtet med Egde: *“Ikke tenk EgdeAI (en av Egdes tjenester), men heller AI hos Egde!”*. Kort tid etter dette møtet ble det formulert et prosjektforslag for hvordan vi i Gruppe 9 kunne bidra til denne visjonen.

LearnEgdeAI – AI for læring og kunnskapsutvikling

Etter samtale med Øyvind Brekkhus Sandåker ved Egde, 10. November 2023, har gruppe 9 kommet frem til et prosjektforslag for vårsemesteret 2024 – LearnEgdeAI.

For å bygge videre på det nye, voksende leddet i Egdes fagområder, AI i Egde, ønsker vi å bidra til arbeidet gjennom et bachelorprosjekt som bygger på tanker og ideer fra det eksisterende systemet EgdeAI.

Vi introduserer – **LearnEgdeAI** (tittel WIP). Et AI-system med mål om å effektivisere nyansettelses prosesser, onboarding og generell læring ved Egdes mange prosjektteams. Det vil si at et prosjektteam i Egde vil ha sin egen lærings AI-Coach som kan effektivt hjelpe med å svare på spørsmål rundt teamets arbeid, teknologier, kunder og partnere.

Grunnlaget for dette er å bidra til å spare interne tidsressurser og effektivisering ved nyansettelser og kunnskapsutvikling i bedriften. Hvis en person står overfor utfordringer med en oppgave eller ønsker å lære om et spesifikt fagområde, kan vedkommende benytte seg av AI-verktøyet for å søke etter relevant kunnskap. En ny og nysgjerrig ansatt kan også benytte seg av verktøyet for egen læring utenfor arbeidstid, om ønskelig. Målet er å redusere tidsbruken som ellers ville gått med til lange læringsvideoer på internett og omfattende lesing av kodeforum og lignende. Dette kan AI-en gjøre for den ansatte og oppsummere informasjonen på en lettfattelig måte.

Med mål om å teste nye teknologier og ressurser er ønsket å skape en kunnskapsbase basert på dokumentasjon fra diverse relevante teknologier ved hjelp av f.eks. LangChains WebBaseLoader (for online dokumentasjon), PDF-document loader (for registrering av PDF data i kunnskapsbasen), Youtube transcript (fra video dokumentasjon fra trygge YouTube-kilder) og lignende komponenter.

Case eksempel hos Egde #1:

Å starte som ny ansatt i en bedrift og/eller prosjektteam kan være både spennende og utfordrende. Ved hjelp av LearnEgdeAI, kan en nyansatt effektivt finne svar på sine spørsmål, relatert til prosjektteamet, ved hjelp av spørringer til systemet.

- Hvilke teknologier bør jeg lære meg?
- Hvor kan jeg tilnærme meg kunnskap om [...]?
- Etc.

Case eksempel hos Egde #2:

Som en allerede etablert Egde ansatt, kan det være av god interesse å lære mer om systemene som brukes i ditt prosjektteam. Ved hjelp av LearnEgdeAI kan du effektivt spørre etter relevant informasjon for dine prosjekter og teknologier, som for eksempel teknologi dokumentasjon og lignende.

- Hvordan kan jeg effektivt sette opp et system for [...]?
- Hva er "best practise" for føring av akkurat dette ifølge dokumentasjonen?
- Etc.

LearnEgdeAI som produkt hos kunder

Dette er også et produkt som potensielt kan bli tatt i bruk av andre bedrifter. Ved å bygge på et etablert rammeverk, er det bare kunnskapsbasen som skiller hva slags bedrift som kan ta dette i bruk. Nyansettelse og opplæring kan ta lang tid og kreve mye ressurser. Et produkt som dette kan potensielt erstatte en god del av tiden som ansatte hos en bedrift bruker på å lære opp nyansatte. Samtidig kan man også bruke produktet dersom noe har gått i glemmeboken, eller om man skal få nye arbeidsoppgaver i samme bedrift.

Eksempler på hvor disse prinsippene kan brukes hos potensielle kunder:

- **Dyreparken i Kristiansand:** Ved onboarding av nye sommervikarer kan et slikt verktøy effektivt hjelpe de nye vikarene 24/7 med spørsmål relatert til sin nye arbeidsplass (for eksempel info om spesifikke dyr, kioskarbeid, policyer, etc.). Kunden velger selv hva slags informasjon som er relevant for systemet, og LearnEgdeAI svarer basert på den bestemte kunnskapsbasen.
- **Instrukser og informasjon:** Har din bedrift en rekke instruksjoner og dokumentasjon i form av PDF-filer og eller utspredd dokumentasjon, kan LearnEgdeAI samle all nyttig

informasjon på en plass. Ett sted for å finne all nødvendig informasjon via din personlige AI-coach, ferdig lagret i kunnskapsbasen.

- **Automatisk språkoversettelse ved samtaler:** når en bedrift ansetter utenlandske arbeidere kan AI-verktøyet ta den norske opplæringsfilen og oversette til morsmål for arbeiderne. Dette kan spare arbeidsgivers tid, da de ikke trenger å måtte utvikle samme opplæringsdokument på flere språk. I tillegg til dette slipper arbeiderne å måtte enten forholde seg til et norsk dokument eller et engelsk dokument. Dette kan forhindre problemer som tolkning av opplæringen dersom det er feil i språket hos den ene parten.
- **Automatiserte øvingsoppgaver:** I stedet for å måtte ta i bruk produkter på nettet som koster penger, altså kurs og lignende for å utvide kunnskap innenfor et fagfelt, kan AI'en lage oppgaver basert på kunnskap hos vedkommende. Dette kan føre til økt læringskurve og mer interaktiv læring som både gir vedkommende kunnskap innenfor problemløsning selv, men og læring via tips fra AI'en. En case kan være at en bruker spør om å få lære språket React, så vil AI'en først introdusere hva React går ut på. Brukeren kan stille spørsmål om forskjellige funksjoner og få rask tilbakemelding uten å måtte søke opp hva de forskjellige komponentene betyr. For eksempel kan AI'en levere ut eksempler på koder hvor det finnes feil hvor brukeren må si i fra hva som er feil. AI'en kan komme med tips og svar om ønsket. Interaktiv og effektiv læring.

Prosjektet og samarbeid oppsummert:

Vi i gruppe 9 ønsker å utforske bruk av nye, spennende AI-verktøy for å potensielt effektivisere hvordan en bedrift og deres ansatte tilegner seg informasjon og kunnskap i arbeidshverdagen.

De to viktigste målene ved dette prosjektet omhandler følgende:

- Gruppe 9s egne læringsutbytte ved prosjektarbeidet. Prosjektet byr på mye ny læring av ukjente systemer, komponenter, prosesser og teknologier som kan styrke gruppens og hvert individs kompetanse innen fagfeltet, samtidig som man videreutvikler allerede etablert erfaring og kompetanse.
- At arbeidet resulterer i et produkt/rammeverk som gir verdi til oppdragsgiver/prosjektkunde. At systemet kan bidra til effektivisering av læring for nye og eksisterende ansatte ved en bedrift som tar i bruk produktet. Ikke minst at resultatet er en del av videreføringen og utviklingen til AI som fagfelt i Egde.

Plan for prosjektgjennomføring

Som beskrevet i tidligere oppgaver i faget IS-310 (som også kan lese om på [vår nettside](#)), har vi i gruppe 9 erfaring med prosjektarbeid gjennom flere semestre ved UiA. Viktige aspekter ved vårt prosjektarbeid ligger ved fokuset på kvalitet, effektivitet, åpenhet og kommunikasjon.

Vi har alltid hatt en sterk tro på at grundig planlegging, regelmessig og kontinuerlig evaluering, samt tett samarbeid både internt i teamet og med produktansvarlig, øker sannsynligheten for at et prosjekt blir vellykket. Dette er prinsipper vi har erfart og utarbeidet ved prosjektarbeid med kunder som Nordic Door, Vitensenteret og andre prosjektoppgaver ved UiA de siste 3 årene. Disse tankene og prinsippene blir desto viktigere ved det kommende bachelorprosjektet våren 2024.

Prosjektomfang

Det forventede totale arbeidsomfanget kan kun spekuleres på dette tidspunktet. Det er forventet at det totale arbeidsomfanget med tanke på utarbeidelse av kompetanse, produkt, rapportering og

dokumentasjon vil kreve tilsvarende arbeidstimer som er tilrettelagt for et fullt vårsemester. Kontaktperson ved Egde, Øyvind Brekku Sandåker, nevnte at basert på han sin erfaring er det viktig å ikke overestimere tilgjengelig tid av for eksempel utviklingstimer da mye tid blir reservert til akademisk skriving og rapportering.

Vi forventer et relativt fullt prosjektløp fra start til slutt. Det vil si at det blir nødvendig å opparbeide en produktforståelse/ønske via innhenting av informasjon og analysearbeid. Da forslaget er å basere dette prosjektet på den eksisterende tjenesten EgdeAI, er det avgjørende å opparbeide en forståelse for systemets funksjonalitet og sammensetning for å best mulig etablerer tilsvarende løsning ved det planlagte, nye systemet. Dette vil kreve forståelse for nye kodestrukturer, språk og tjenester som per dags dato er ukjente for majoriteten av prosjektgruppen. Det er derfor viktig å avsette tid til effektiv research og kompetanseutvikling i starten av prosjektgjennomføringen.

Det som generelt er en fellesnevner for forståelsen for prosjektomfanget, ligger ved å oppnå en felles forståelse og utarbeide en backlogg med realistiske tidsestimater og oppfølging for å garantere at prosjektet holder forventet kurs fra start til slutt. Dette vil enklere kun etableres og defineres når vi starter prosjektarbeidet våren 2024.

Produktfunksjonalitet

Den ønskede funksjonaliteten til sluttproduktet blir beskrevet mer i detalj i prosjektbeskrivelsen over. Nøkkelpunktene som kan nevnes ut ifra den tidligere beskrivelsen er å skape funksjonalitet rundt et KI-basert arbeids-coach/mentor ved en arbeidsplass. Ved å stille spørsmål til tjenesten skal den effektivt kunne hente ut og videreformidle ønsket kunnskap fra en forhåndsbestemt kunnskapsbase. For å unngå repetisjon, kan du lese mer om dette i prosjektbeskrivelsen.

Kompetansebehov, kompetansegap ved prosjektarbeidet

Som tidligere nevnt har vi i Gruppe 9 utviklet bred kompetanse innen prosjektgjennomføring i løpet av de siste fem bachelorsemestrene ved UiA. I vår gruppe har vi medlemmer med interesser innen UX, front- og backend-utvikling, ledelse og kunstig intelligens (KI). Et av gruppemedlemmene har også bidratt til å pionere EgdeAI-tjenesten, noe som bidrar til å heve kompetansenivået for det kommende arbeidet.

Siden det forventes at vi gjennomfører et fullt prosjektløp basert på EgdeAI-tjenesten, er det avgjørende at vi kan utnytte vår kunnskap om prosjektgjennomføring fra start til slutt. Et av hovedmålene med prosjektet er selvfølgelig å ta i bruk og lære nye spennende verktøy som kan realisere ønsket funksjonalitet i produktet. Dette krever imidlertid også at gruppen utvikler den nødvendige kompetansen som kreves for å skape verdi med verktøyene og tjenestene. Selv om flere av medlemmene i gruppen har interesseområder som reflekterer flere av disse verktøyene og tjenestene, er det viktig å understreke hvor avgjørende det faktisk er at kompetansegapet fylles der det er påkrevd og nødvendig for å realisere vårt ønskede mål for prosjektet og sluttproduktet.

Kvalitet og QA

For å sikre kvalitet i prosjektet, anser vi kvalitetssikring som en essensiell del av arbeidsprosessen. Målet vårt er jevnlig å levere produkter og prosjekter av høy kvalitet som oppfyller kravene og ønskene til vår samarbeidspartner, samtidig som de samsvarer med våre egne standarder. Vi vektlegger betydningen av å være stolte av det utførte arbeidet og at overleveringsprosessen skaper verdi for alle involverte interessenter.

Et prosjekt av god kvalitet for oss er et prosjekt der produktet eller tjenesten som blir utviklet i løpet av prosjektløpet, oppfyller kravene og forventningene fra både kunden og arbeidsgiver. Selv om dette kan virke som en selvfølge, krever veien til et slikt kvalitetsprodukt grundig struktur og

pålitelig kvalitetssikring gjennom hele prosjektløpet. For å oppnå dette resultatet har vi god erfaring med å aktivt og hyppig teste arbeidet både internt i gruppen og sammen med interessenter og en eventuell målgruppe. Vi arrangerer også jevnlig interne demo- og expo-workshops for å engasjere alle gruppemedlemmene i prosjektarbeidet. Videre gjennomfører vi ulike tester, inkludert brukertesting og enhetstesting. For ytterligere kvalitetssikring av programmeringsprosjekter benytter vi også metoder som code review.

For å styrke vår vellykkede tilnærming til prosjektarbeid, vurderer vi følgende forbedringer:

- **Utvikling av Interne Demo/Expo Workshops:** Vi vil utvide bruken av interne demo- og expo-workshops for bredere deltakelse og økt informasjonsdeling blant gruppemedlemmene og prosjektets interessenter.
- **Optimalisering av Teststrategi:** Vi fortsetter å forbedre vår teststrategi, spesielt innen brukertesting og enhetstesting, for å identifisere potensielle problemer tidlig i utviklingsprosessen.
- **Forbedring av Code Review-prosessen:** Vi planlegger å optimalisere vår code review-prosess med tydelige retningslinjer og beste praksiser for å styrke kvalitetssikringen i programmeringsprosjekter.

Disse tiltakene sikrer mot å opprettholde og forbedre vår forpliktelse til å levere høykvalitetsprodukter, samtidig som vi aktivt involverer alle teammedlemmer i prosessene.

Samarbeid, arbeidsdeling, arbeidsrutiner.

Gjennom kontinuerlig innsats på universitetet har vår gruppe utviklet solid samarbeidserfaring og kompetanse ved å håndtere forskjellige prosjekter og oppgaver. Denne kontinuerlige innsatsen har resultert i etableringen av en stabil gruppedynamikk, hvor vi har grundig innsikt i hverandres kompetanseområder, styrker og forbedringspotensialer. Med en dedikert scrum-master/ordstyrer tar gruppen ansvar for den nødvendige administrasjonen og oppfølgingen i våre arbeidsprosesser.

Vi har solid erfaring med å jobbe innenfor den agile arbeidsmetodikken Scrum, og har utviklet betydelig kompetanse ved å benytte kanban boards i GitHub Projects, Trello og DevOps for å organisere arbeidet. Etter grundige tester har vi konkludert med at GitHub Projects og DevOps er de optimale plattformene for oss og vårt arbeid. Begge gir oss tilgang til varierte funksjonaliteter som vi drar nytte av i prosjektarbeidet. Ved hjelp av en nøye utarbeidet backlog, fordeles oppgaver og ansvar enkelt blant gruppemedlemmene basert på individuell kompetanse, erfaring og læringsmål. På denne måten maksimerer vi effektiviteten til gruppen.

Selv om vi har utviklet et godt vennskap internt i gruppen, har vi alltid sett viktigheten av å kunne skille mellom vennskap og arbeid. Det er viktig å ha et sunt, positivt arbeidsmiljø hvor man kan dele ideer åpent, gi konstruktive tilbakemeldinger og samarbeide effektivt. Denne profesjonelle tilnærmingen bidrar til å skape et klima hvor alle gruppemedlemmer føler seg komfortable med å uttrykke sine synspunkter og bidra til prosjektet på en konstruktiv måte. Vi verdsetter derfor tydelige kommunikasjonskanaler, transparente forventninger og et gjensidig ansvar for å opprettholde et produktivt og trivelig samarbeidsmiljø.

Avslutning

Vi avslutter faget IS-310 - Prosjektgjennomføring med stor entusiasme, vel vitende om at dette markerer starten på slutten av vår nåværende akademiske bachelor opphold ved Universitet i Agder. Vårt kommende bachelorarbeid, planlagt for våren 2024, står som avslutningen på seks

intense semester ved UiA og har potensial til å være springbrettet inn i arbeidslivet. Samarbeidet med Egde for prosjektet '**LearnEgdeAI**' representerer ikke bare en unik mulighet for oss som gruppe, men også en sjanse til å skape verdi og innovasjon innen bruk av kunstig intelligens.

Prosjektet tar sikte på å utvikle LearnEgdeAI, et AI-system designet for å effektivisere nyansettelser, onboarding og generell læring ved Egde. Vi ser frem til å utforske nye teknologier og ressurser, samtidig som vi bygger på vår allerede etablerte erfaring. Gjennom grundig planlegging, kontinuerlig evaluering og tett samarbeid, er vi forberedt på å håndtere det forventede arbeidsomfanget og fylle eventuelle kompetansegap som måtte oppstå.

For å sikre kvalitet i prosjektet, vektlegger vi kvalitetssikring som en essensiell del av arbeidsprosessen. Våre tiltak inkluderer utvidelse av interne demo- og expo-workshops, optimalisering av teststrategi og forbedring av code review-prosessen. Vi er fast besluttet på å opprettholde og forbedre vår forpliktelse til å levere høykvalitetsprodukter samtidig som vi aktivt involverer alle teammedlemmer i prosessene.

Gjennom vårt sterke samarbeid, dedikerte arbeidsrutiner og fokus på profesjonalitet, ser vi frem til å levere et vellykket prosjekt med betydelig verdi både for gruppen og Egde.

Dette representerer ikke bare avslutningen på vårt nåværende studieprogram, men også starten på en spennende fase der vi får anvende og videreutvikle vår kunnskap gjennom praktisk samarbeid med en ekte partner innen kunstig intelligens og innovasjon.

Litteraturliste:

Egde. (u.å.). *Vi forener mennesker og teknologi*. Hentet 28.11.23 fra <https://egde.no/vi-jobber-med/>

Vedlegg 5 - Sprint Planning

Sprint Planning – Sprint #5

Teammedlemmers oppmøte:

- Frida
- HC
- Hassan
- Jostein
- Michael

Notater fra møtet:

1. Sprint Planning diskusjon:

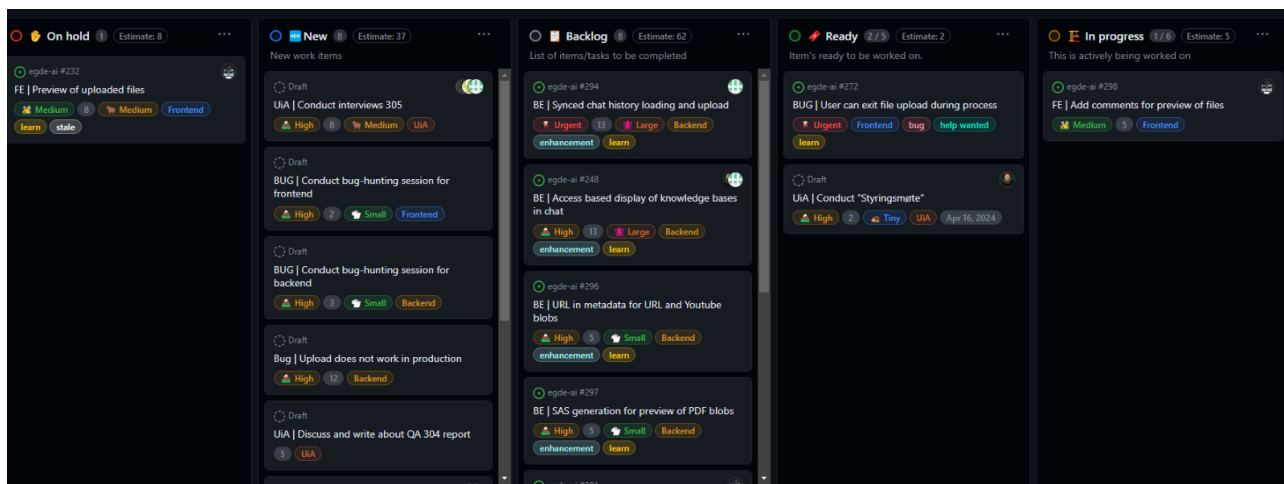
- Stadig nærmere vi oss slutten på bachelorprosjektet. Sprint #5 markere en viktig fase av bachelorprosjektet og systemutviklingen, for det er nemlig på slutten av sprinten av “code-freeze” tar effekt. Det vil si at all form for utviklingsarbeid blir stoppet frem til bachelorleveringen 16. Mai.

De mest relevante datoene for dette er:

- *Soft-freeze* – 26. April
- *Hard-freeze* – 6. Mai
- I møtet kom vi frem til at Sprint #5 skal i hovedsak omhandle følgende gjøremål og milepæler.
 - Fullføre IS-305
 - Code-blocks i chat-respons
 - Chat-historikk
 - Flere guard-rails i admin page
 - Fullført CR(U)D operasjoner i admin page
 - Bug-hunting
- Den generelle vurderingen blant gruppen er at vi ligger godt an til å komme i mål med våre ønsker for prosjektet før deadline for hard-freeze slår inn 6. Mai. Det er noen såre punkter, spesielt innen opplastning mulighetene, som bidrar til en svak grad av skeptisisme. Dette er hovedsakelig grunnet problemer med noen av pakkene som blir brukt i systemet, og løsningen ligger utenfor våre hender ved mindre vi bytter pakkestruktur som vil ta lengre tid. Noen problemer ved PDF-opplastning i produksjon skaper også problemer, men som letter skal kunne justeres og fikses.

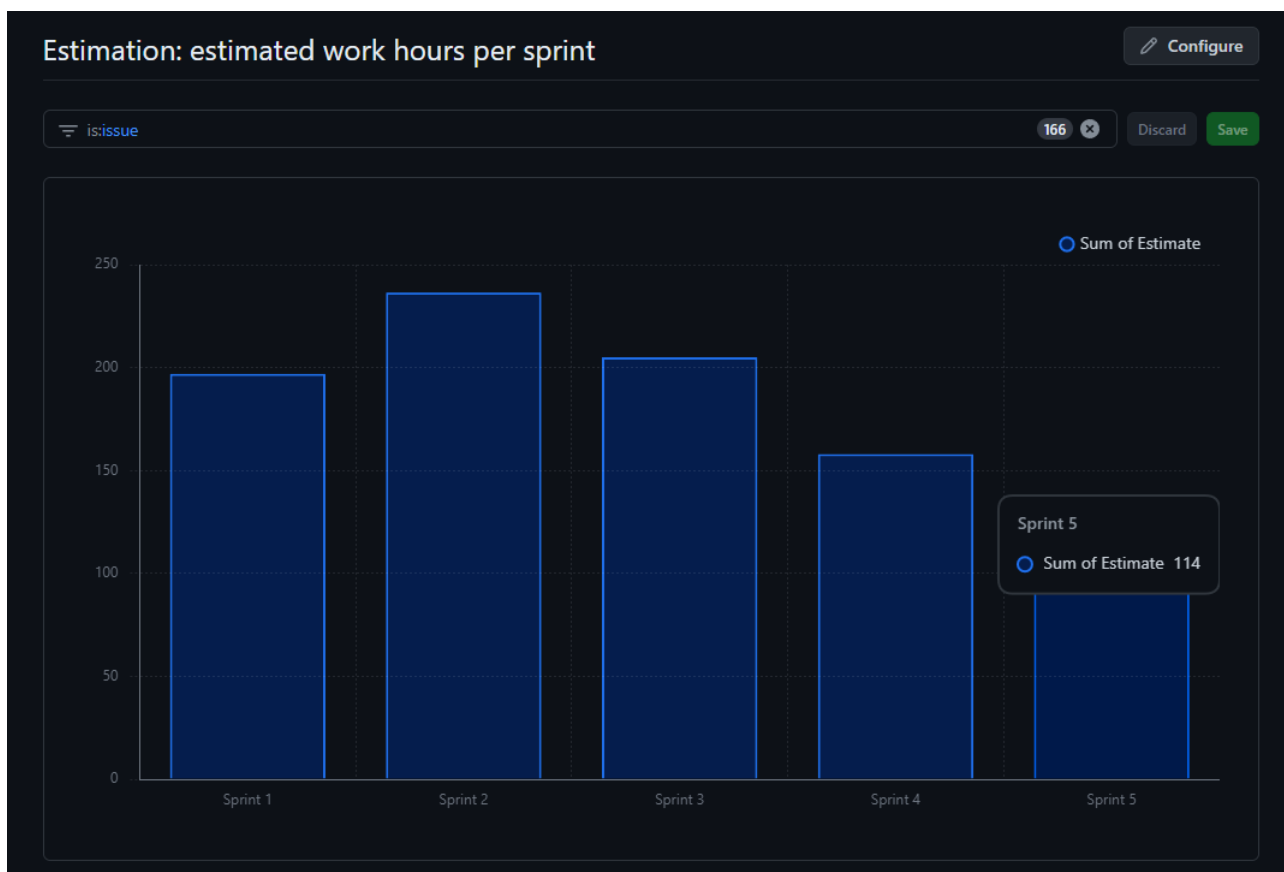
2. Backlogg planning:

- Planleggingen starter med å se kjapt over de allerede tilgjengelige artiklene fra forrige sprint. Disse har vi allerede brukt mer tid på ved sprint #4 retrospective, så vi går raskt videre til neste del.
- Som vanlig settes det av 15 minutter til en rask økt for oppretting av de mest sentrale backlogg-artiklene for den nye sprinten. Disse baserer seg på milepælene som blir nevnt over. Totalt sett starter sprint #5 med 20 issues/backlogg-artikler.



3. Planning poker for tids estimering

- Planning poker blir igangsatt, hvor hvert av de nye og gamle issueene blir vurdert basert på hvor lang tid det estimeres å gjennomføre arbeidet fra start til slutt. 114 arbeidstimer blir estimert sammenlagt for sprint #5.



4. Sprint planning avslutning

- Avslutningsvis tas en felles gjennomgang av Miro modellen for sprint #4 retrospective analysert og påminnet om for arbeidet videre.

Vedlegg 6 - Daily Scrum

Gruppe 9 - Daily Stand-up logg

Scrum master: Michael Tangeraas

Dato: Tirsdag 05.03.24

Teammedlemmer til stede:

- Jostein
- Frida
- Hassan
- Michael
- HC

Dagsorden for møte:

1. Hva oppnådde du i går?

- **Frida:** Lagde skisser og wireframes for data oversikt (altså når man klikker på en artikkel som er lastet opp). Fikk startet litt på prototype.
- **Jostein:** Fikk lagt til nødvendige, manglende knapper for opplasting, frontend muligheter for opplasting av flere artikler, modal for sletting av file/filer.
- **Hassan:** Fikk på plass filter funksjonalitet i admin dashboard. Dette lar en bruker filtrere gjennom kunnskapsbaser o.l.
- **HC:** Fikset litt på user response ved opplasting av filer. Meldingene går ikke veldig i detalj, men det er egne meldinger for hver feilmelding. Så litt på sletting av namespace fra Pinecone.
- **Michael:** La ut workplace-oppdatering for student progressjon, oppdaterte backlogg-item size schematic i dokumentasjon, oppdaterte nye backlogg-items, lagde ny dokumentasjon for kode-struktur.

2. Hva planlegger du å gjøre i dag?

- **Frida:** Skal fortsette med prototype av data oversikt.
- **Jostein:** Prøve å liste ut filer for sletting, begynne å se på å lage komponent for AI hallucination warning.
- **Hassan:** Fikse issueet med ShadCN warning i nåværende løsning. Gå videre på tilgjengelige backlogg artikler.
- **HC:** Sletting i Blob-storage, vektor-ID basert sletting i Pinecone.
- **Michael:** Prøve å ferdigstille første draft av userguides (endelig). Som en del av den oppgaven skal jeg prøve å bli litt klok på best-practice for komponent sortering i React.

3. Er det noen hindringer eller utfordringer i veien?

- **Frida:** All's good
- **Jostein:** All's good
- **Hassan:** All's good
- **HC:** All's good
- **Michael:** All's good

4. Ekstra notater:

-

Vedlegg 7 - Sprint Retrospective

Sprint retrospective – Sprint #4

Teammedlemmers oppmøte:

- Frida
- HC
- Hassan
- Jostein
- Michael

Notater fra møtet:

Sprint #4 oppsummert:

1. **Backlogg** – Gjennomgangen av backloggen ble utført i lik stil som vanlig. Møtet starter med en gjennomgang av alle backlogg-artikler fra sprinten. Hver forfatter/assignee forklarer kort hva som har skjedd i arbeidet og resultatet av backlogg-artikkelen.

Det var estimert **158** arbeidstimer for denne sprinten, og vi avsluttet med **101** registrerte arbeidstimer. Hovedgrunnen til den store forskjellen i i estimeringen og resultat kommer blant annet av at noen av backlogg-artiklene ble ført som “closed as not planned”. En av disse issueene hadde en estimert arbeidstid på 21 timer med null registrerte arbeidstimer, noe som gjør at det totale utslaget mellom estimert og registrert blir ganske høyt.

Backlogg-artiklene som vi ville videreføre eller som ikke var ferdigstilt ble flyttet over til neste sprint, og artiklene som ikke var ferdig utfylt ble ferdigstilt og behandlet deretter.

2. **Refleksjon – Positive tanker fra Sprint #4** – I denne iterasjonen av Sprint-review valgte gruppen å gjennomføre en workshop med 3-steps retrospective refleksjon. Her fikk vært medlem mulighet til å skrive opp punkter for hva som gikk bra, hva kunne vært bedre, og hvilke handlinger vi ønsker å ta med inn i neste sprint. Under finner du punkter fra de viktigste funnene for hvert av temaene.
 - **MoSCoW og Benyons:** Det var planlagt å gjennomføre en revurdering og analyse av dagens løsning opp mot både MoSCoW og Benyons vurderingene våre. Begge møtene ble gjennomført og ga god innsikt i hva som blir gjort bra og ha som må endres.
 - **Rapportskriving:** Vi har viet mye mer tid til rapportskrivning denne sprinten og det kan man lett se på sidetall og ordtelling. Dokumentet begynner å bli ganske stort og mye godt innhold. Vi var tidlig bestemt på at vi skulle slippe et “skippertak” scenario i mai, og vi er nå på god vei til å slippe dette.

- **Samarbeid og kaffeprat:** Ikke overraskende fortsetter det gode samarbeide i og på tvers av arbeidsteamene. Noe føler også at vi har vært enda flinkere på å bruke hverandre enn tidligere. Kaffepraten og distraksjoner har det også blitt færre av, og har eventuelt blitt flyttet til kaffeplassen.
- **Backlogg og tids estimering:** Det kommenteres at tids estimeringen vår er gode, både når det kommer til selve føringen av timer og å treffe innenfor rammene som ble forhåndsbestemt.



3. Refleksjon – Hva kan vi bli bedre på i neste sprint - tanker fra Sprint #4.

- **IS-305 rapport:** Rapporten i IS-305 har “ulmet” over oss en liten stund, og det er på tide å ta et tak for å bli ferdig med den. Vi ligger veldig godt an i forhold til deadline, men tror motivasjonen for å gjennomføre den har vært litt laber blant alle gruppe medlemmene.
- **Føring av backlogg:** Vi har kontinuerlig blitt bedre på å føre backloggen gjennom alle sprintene, men det er fortsatt smårusk som må fikses i og ting som blir glemt. Alle er innstilt på at dette må bli tatt mer seriøst.
- **Kvalitetssikring og testing:** I det siste har flere bugs og uferdig funksjonalitet blitt pushet for tidlig opp i produksjon. Dette skyldes blant annet av at noen av disse feilene ikke kan bli fanget i utvikling grunnet Azure miljøet for development og produksjon som er uavhengige av hverandre. Det bør også poengteres at man har nok vært litt raske og ikke testet funksjonaliteten nøye nok før det har blitt pushet til *main*. Dette er selvfølgelig en av svakhetene med å ha en løstere tilnærming til *main* branchen og hva som kan legges der. Vi har som nevnt tidligere praktisert en rimelig agil tilnærming der vi ofte og raskt får ut ny funksjonalitet, men i det siste har det kanskje gått litt for fort.
- **Dokumentasjon i kode:** en gjenganger som nå skal bli tatt mer seriøst. Vi MÅ bli flinkere på å dokumentere i koden vår.

2

What can we do better?

305 rapport!!

Kvalitets sikring

Ting må testes mer i prod (vercel)

Bugs/feil må føres opp

Stress teste systemet for bugs

Fortsatt lite kommentarer

Bli enda flinkere på å oppdatere backlogg 🙌

Skrive tester

Bug teste funksjon teter 🚗

Kommentar i koden

Skrive innledning på backlogg

Refleksere på testing av kode/Dokumentere

Segmentere kode. VELDIG store filer om dagen 📁

Mer comments i kode 🗨️ 🗨️

Description on Backlogg 📄

finne på nokke sosialt

Sosial samling 🗺️

bruke sofaen enda meir

Finne tid og engasjere til sosial aktiviteter 🏆

4. Til neste sprint – Hva vil vi ta med oss i arbeidet videre?

- **Innkommende code-freeze:** Flere kommenterer og forbereder seg på den innkommende *code-freezen* som vil medføre at videreføring av utvikling i kode vil bli delvis eller helt stoppet nærmere slutten av april og starten av mai.
- **Kvalitet i koden:** Relatert til kommentaren for forbedringsforslag, så skal vi bli enda flinkere på å dokumentere koden bedre. Dette gjøres for å gjøre det lettere for seg selv og andre å lese og forstå koden.
- **Workshop/Code-review:** Vi rakk ikke å gjennomføre workshop forrige sprint grunnet påsken, men det betyr ikke at vi ikke vil ha det i sprint #5! Vi skal booke inn til workshop i midten av sprinten.
- **Sosialt med gruppen:** Det ble også nevnt under hva vi kan gjøre bedre at det var ønskelig med flere sosiale-samlinger, og dette skal vi få til. Jostein har allerede booket oss inn til quiz på Kraftverket under første uken av sprint #5.
- **“Ikke stoppe å løpe før etter mållinja!”:** Vi nærmer oss deadline og slutten på bachelorprosjektet for denne gang, men vi må huske å holde motivasjonen og effektiviteten gående som vi pleier helt fram til slutt.

3

Actions for next sprint

Ikke stoppe å løpe før mållinja 🏃

Ikke opprette nye backlog items (Soft freeze)

Se over prosjektet og klargjøre det for freeze 🤖

Forberede til code freeze 🤖

Sosialt utenfor jobb 🗣️ (Jostein mekker)

delta aktivt i møte

Vi skal sjekke backlogg om morgenen 🕒

Fullføre IS-305 🍌

Få på plass/prioritere alle "must" i prosjektet

Kvalitet over kvantitet

Fin pusse all kode i prosjektet 🍌

Segmentere kode der vi kan

Skrive kode kommentar 🍌

Vi må kommentere mer

Komme enda lengre på rapporten 📄

felles gjennomgang / workshop

Gå gjennom og forstå koden

Forstå kode

Code review

Avslutnings workshop på slutten av sprint for å gå igjennom alt sammen

Vedlegg 8 - Ukeslogg 10

Logg uke 10

Produsert av Gruppe 9 – Bachelorprosjekt ved Egde Consulting.

Skrevet av Scrum master – Ivar Michael Tangeraas

Mål for uken:

1. Sprint planning og oppstart av Sprint #3.
2. Fortsette brukertester av Learn brukergrensesnitt. Kanskje mer fokus på admin-page?
3. Fortsette på implementasjonen av admin-page i utvikling.

Ukesoppsummering:

Uke 10 markere starten på Sprint #3. Hvor sprint #2 hadde som mål om å kunne presentere en tidlig demo for Learn tjenesten, har sprint #3 fokus på å forbedre de implementasjonen som ble påbegynt i sprint #2. Dette gjelder spesielt funksjonaliteten som omringer det mer administrative brukergrensesnittet for administratorer og kommende *kunnskapseiere*.

Hva er en *kunnskapseier*? I konteksten av Learn er det påtenkt at de forskjellige kunnskapsbasene en bedrift ønsker å opprette, bør ha en eller flere personer som kan administrere og å delegerer nødvendige data etter behov. Et viktig aspekt med Learn som tjeneste er at responsen fra tjenesten er kun like god som dataene den har tilgang på. Det vil si at en kunnskapsbase med mye relatert og oppdatert data vil kunne komme med gode responser til en brukers spørsmål, enn en mager, utdatert kunnskapsbase vil. Derfor har vi påtenkt at kunnskapsbasene burde ha en eller flere ansvarspersoner som kan stå i rollen for å sikre kunnskapsbasens styrke. Denne rollen kan ha både stor og liten omfatning, basert på hvor ofte dataene trenger å bli oppdatert eller hvor mye data som må inn første gang. I Egdes tilfelle kan det være interessant å lage kunnskapsbaser for hvert av teamene i Smia. Da vil det være naturlig at interessenter i teamene vet mest om teamet og hvilke data som burde ligge i deres kunnskapsbaser. Om dataene blir oppdatert ofte vil det være smart om vi setter om rammer for å bidra til dette, som for eksempel *Cron jobs*.

Tilbake til ukensoppsummering, uken startet med et sprint planning møte hvor vi sammen gikk gjennom forventninger for sprinten, hvilke backlogg-artikler som fulgte over fra Sprint #3 og eventuelt hvilke nye artikler som burde opprettes. Planning poker ble gjennomført på eksiterende og nye artikler, og arbeidet startet som vanlig når møtet var over.

Gjennom uken har agendaen hovedsakelig vært fokusert på følgende elementer:

- Legge ut en workplace oppdatering om Learn
- Forberedning på A/B-tester i UX-teamet.
- Implementasjon av user guides i produksjon
- En rekke oppdateringer og bug-fixes i frontend
- Funksjonalitet for å velge kunnskapsbase ved opplasting av data
- Begynt arbeid for å trygt slette data fra kunnskapsbasene
- Og andre småting.

Refleksjon:

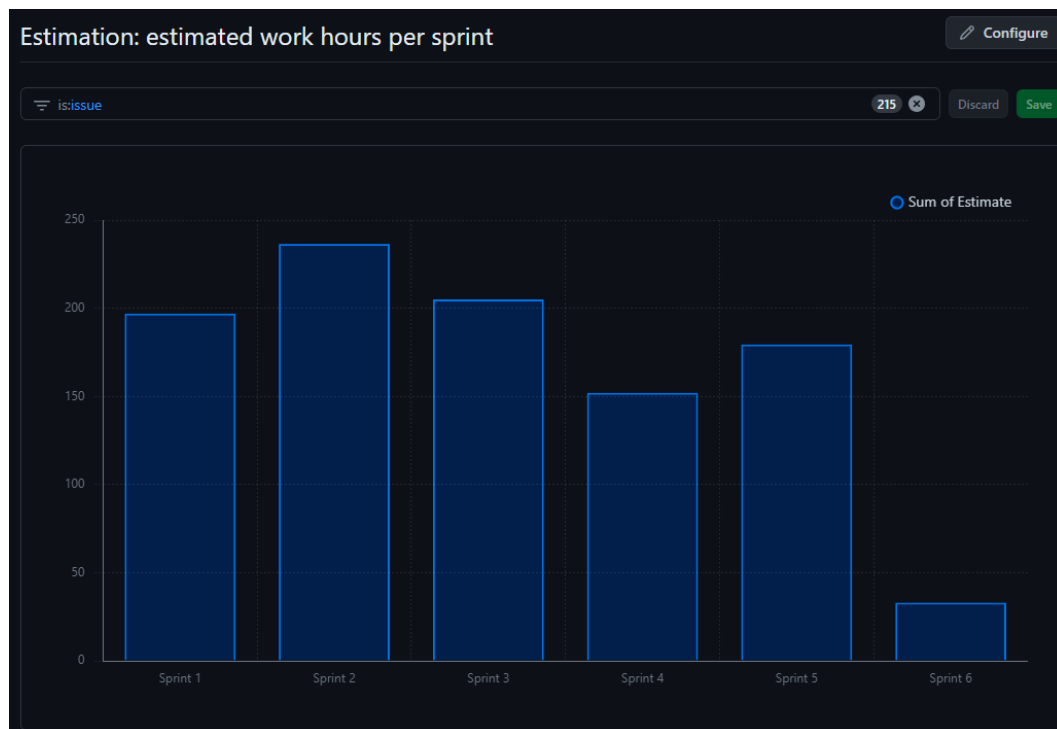
- **Hva gikk bra?**
 - Sprint planning møtet gikk som forventet og gruppen hadde en felles forståelse og ønske for hva som måtte prioriteres i sprinten.
 - Arbeidet har fortsatt med mindre problemer.
 - Gjennomførte et lite sosialt arrangement med taco og filmkveld i løpet av uken.
- **Hva var vanskelig?**

- Noen backlogg-artikler hadde i mindre grad feil-estimert forventede arbeidstimer da problem scopet var større en forventet. Med litt overtidsarbeid kom vi fortsatt i mål med godt resultat.
- **Noen uventede utfordringer?**
 - Vi oppdaget en intern server-error i produksjonen av Learn tjenesten som ikke matchet resultat i development. Etter en rask og effektiv diskusjon kom vi frem til at det var manglende/utdaterte environment variabler i Vercel som måtte oppdateres for å møte dagens produksjon.

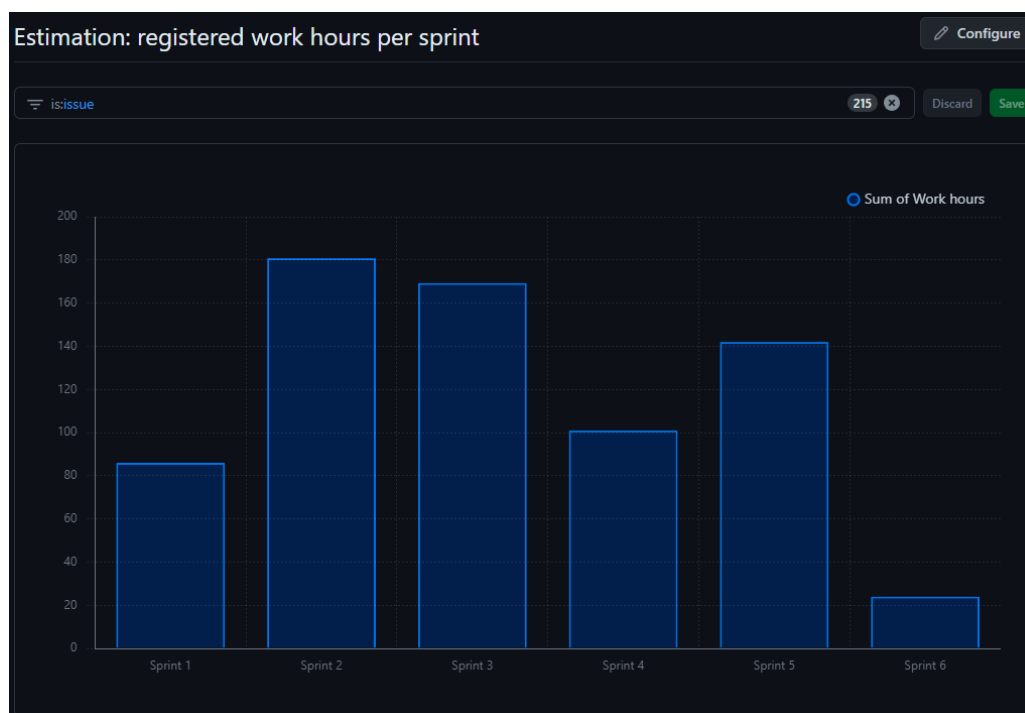
Neste ukes mål:

1. Fortsette arbeidet rundt administrator funksjonaliteter
2. Starte implementasjonen av "guided tour" systemer for Learn
3. Begynne å gjennomføre A/B-tester

Vedlegg 9 - Estimeringsgraf #1 – Estimert arbeidstid per Sprint



Vedlegg 10 - Estimeringsgraf #2 – Registrert arbeidstid per Sprint



Vedlegg 11 – Risikoanalyse

Risikoanalyse - Bachelorprosjekt

Hendelse	Sannsynlighet	Konsekvens	Risiko	Beskrivelse + Tiltak
Dårlig samarbeid/kommunikasjon	1	4	4	Dårlig samarbeid og/eller kommunikasjon har en kritisk innflytelse over prosjektet. Dersom bachelorgruppen ikke klarer å jobbe bra sammen, vil minimalt med progresjon bli oppnådd. Godt samarbeid og kommunikasjon er noe som gruppen har opparbeidet gjennom flere prosjekter via gode rammer for kommunikasjon og oppfølging, som gjør det lite sannsynlig at dette vil påvirke prosjektet.
Avvik av utstyr/tjenester	1	4	4	Ved mangler på API-nøkler, database-tjenester, maskinvare, o.l. vil prosjektøpet oppleve vanskeligheter med å oppnå ønskede resultater. Her er det kritisk at ønsker/nødvendigheter er forhånds diskutert med bedrift.
Dårlig oppfølging fra veileder	1	2	2	Bachelorgruppen er tildelt en veileder ved UiA. Veilederen skal bidra til å ivareta god progresjon og arbeidsflyt via veiledning, oppfølging og forslag. Ved eventuell svakere eller dårlig oppfølging opplever bachelorgruppen at arbeidsflyten vil bli mindre påvirket, grunnet at gruppen allerede er godt etablert rundt prosjektarbeid og dynamikk.
Tekniske utfordringer	2	3	6	Tekniske utfordringer er noe som ofte kan skje i et utviklingsprosjekt og kan være både frustrerende og ha en negativ innvirkning på prosjektets progresjon. For å unngå tekniske utfordringer er det lurt å ha god kjennskap til verktøy som tas i bruk, og hvilke ressurser man har tilgjengelige, for eksempel andre gruppe-medlemmer, kollegaer, veiledere og web-ressurser. Jo raskere et problem løses, jo raskere kan progresjonen fortsette.
Ujevnt effektivt arbeid	2	3	6	Ved tilfeller av ujevnt effektivt arbeid vil progresjonen være lavere til tider. Få å unngå lav effektivitet er det essensielt å jobbe på en sunn måte der man jobber strukturert i passende arbeidsperioder. Dette gjør at man klarer å holde den generelle motivasjonen høy og progresjonen jevn. Dette kan oppnås ved å ha rom til pauser, god planlegging, diskusjoner, og sosiale aktiviteter gjennom prosjektøpet.
Mangel på kompetanse	3	3	9	Mangel på kompetanse er noe som mest sannsynlig møter på i et prosjekt der nye og ukjente teknologier skal tas i bruk. For å løse en problemstilling der man har manglende kompetanse er det viktig å sette seg inn i tematikken og lære så mye som mulig om den slik at man kan progressere. Som 3. års bachelorstudenter har vi innarbeidet oss metodikker som gjør at vi klarer å oppnå tilstrekkelig kompetanse på en effektiv måte.
Stagning ved prosjektgjennomføring	3	3	9	Dersom stagning i prosjektgjennomføring skjer, vil verdiful tid gå tapt. For å unngå dette må man tilpasse og fordele ressurser i bachelorgruppen etterhvert som størrelse på arbeidsoppgaver endres. God kommunikasjon rundt arbeidet er også kritisk for å ivareta god flyt. Dette kan blant annet ivaretas gjennom scrum møter og diskusjon.
Feilestimering av ressurser	3	2	6	Å estimere arbeidsressurser tilgjengelig i et prosjektarbeid er viktig for å danne et estimat rundt hva som er mulig å oppnå innenfor predefinerte rammer. Slike feilestimeringer kan føre til tapte muligheter og/eller tap av forhåndsbestemte funksjoner og tjenester. For å motvirke dette er erfaring og realistiske forventninger til ressurser viktig å ta i betraktning for å unngå større feilestimeringer.
Sykdom	3	1	3	Sykdom er naturligvis noe som kan oppstå i et prosjekt. Her vil graden at sykdom har mye å si. Dersom man blir sengeliggende i flere dager mister man en god del arbeidskraft. Hvis man er i stand til å jobbe digitalt er det en god løsning som minimerer konsekvensene av sykdommen. Digitalt arbeid er et alternativ bachelorgruppen er godt vant med å ta i bruk.
Feilestimering av tidsbruk	4	3	12	Feilestimering av tidsbruk er noe bachelorgruppen har bommet på før, og mest sannsynlig kommer til å bomme på igjen. Skalaen av arbeidsoppgaver er noe som kan være greit nok å estimere, men selve tidsbruken kan være utfordrende å estimere på forhånd. Dersom man beregner for lite tid til en arbeidsoppgave må man tilpasse og flytte ressurser slik at denne arbeidsoppgaven kan bli gjennomført, eller nedskalere selve arbeidsoppgaven.
Utviklingsutfordringer	4	3	12	Store deler av prosjektarbeidet tar for seg utfordringer ved et utviklingsperspektiv av en tjeneste/applikasjon. Da bachelorgruppen kommer til å arbeide med nye verktøy, språk og systemer for å oppnå ønsket resultat. Sannsynligheten for å møte utfordringer ved dette arbeidet er svært høy, og kan føre til store konsekvenser ved arbeidet, men dette er forventet. Med et stødig ressursetnettverk tilgjengelig, erfaring for læring og egenskapelig utvikling er dette en risiko og utfordring gruppen er klar for å møte.

Vedlegg 12 - Spørreskjema

Spørsmål 1/10

1. Alder *

- 20-25
- 26-35
- 36-49
- 50+

Section 2

Spørsmål 2/10

2. Fagfelt *

- UX
- Front-end
- Back-end
- Fullstack
- Ledelse
- Økonomi
- Other

Section 3

Spørsmål 3/10

3. Hvor lenge har du jobbet hos Egde? *

Spørsmål 4/10

4. Hvilke ressurser eller støtteverktøy benytter du deg av under innhenting av informasjon, opplæring eller annet? *

- Fadder/Coach
- Workplace
- Teams
- Spør rundt på kontoret
- Egdenett/Sharepoint
- Opplæring
- Other

Section 5

Spørsmål 5/10

5. Hvordan foretrekker du å lære nye ting i arbeidsmiljøet ditt? *

- Lese
- Video
- Kurs
- Muntlig
- Other

Section 6

Spørsmål 6/10

6. På skalaen under, hvordan stiller du deg til bruk av AI-verktøy som hjelpemiddel i ditt arbeid? *



Spørsmål 7/10

 Enable

Turn on "Allow responses" so data before you

7. Vile du benyttet et AI-verktøy til å innhente informasjon fra EgdeNett (Sharepoint)? *

- Ja
- Nei

Section 8

Spørsmål 8/10

8. Om du har brukt EgdeAI, beskriv kort hva som fungerer bra, og hva som fungerer dårlig? *

Enter your answer

Section 9

Spørsmål 9/10

9. Kunne du tenke deg å delta i et fysisk intervju om dine tanker og ideer rundt en AI-verktøy? *

- Ja
- Nei

Spørsmål 10/10

10. Vennligst oppgi noe kontaktinformasjon som vi kan nå deg på (Navn, Teams, Email, Nummer, osv.) *

Enter your answer

11. Alder? *

Enter your answer

12. Fagfelt? *

Enter your answer

13. Plass *

- Fysisk (Kristiansand)
- Digitalt (Teams)

Vedlegg 13 - Personas

**Mads Mikkelsen**

Alder: 32 år - f. 1992

Utdanning: Informatikk

Rolle: Utvikler

Ansiennitet i bedriften: 1/2 år

Erfaring: 9 år

◆ Biografi

- ◆ Mads er en erfaren og kunnskapsrik fullstack-utvikler. Han har på kort tid blitt en viktig del av teamet sitt ved å bringe sin ekspertise innen både frontend og backend utvikling.

◆ Holdning

- ◆ Strukturert og rutinebasert person som liker å holde seg opptatt.
- ◆ Verdsetter gode sosiale forhold med sine medarbeidere.

◆ Mål

- ◆ Ønsker å være en verdifull bidragsyter til teamet, og levere prosjekter med høy kvalitet innen tidsfristene.
- ◆ Har ambisjoner om å være involvert i prosjekter som utfordrer han og gir rom for kreativ problemløsning.

◆ Erfaring

- ◆ Tar aktivt i bruk og tester nye AI-verktøy i søket om å bli mer effektiv i sitt arbeid.

**Thomas Husvig**

Alder: 48 år - f. 1976

Utdanning: Prosjektledelse

Rolle: Prosjektleder

Ansiennitet i bedriften: 7 år

Erfaring: 23 år

◆ Biografi

- ◆ Thomas er en erfaren prosjektleder med over 7 års tjeneste i bedriften. Gjennom dette har han bevist seg som en effektiv leder som kan balansere både taktiske og strategiske aspekter ved prosjekthåndtering.

◆ Holdning

- ◆ Han er resultatorientert og fokuserer på å oppnå definerte mål og levere prosjekter til rett tid og innenfor budsjettet.
- ◆ Opptatt av å engasjere og motivere teamet sitt.

◆ Mål

- ◆ Ønsker å skape et arbeidsmiljø som inspirerer til samarbeid hvor teamet trives og yter sitt beste.
- ◆ Ønsker å videreutvikle sine kunnskaper ved å delta på relevante kurs og workshops.

◆ Erfaring

- ◆ Interessert men skeptisk til bruk av AI-verktøy. Har tatt i bruk noen etter anbefalinger fra kollegaer.

**Mariann Sæther**

Alder: 56 år - f. 1968

Utdanning: Økonomi og administrasjon

Rolle: Regnskapsfører

Ansiennitet i bedriften: 6 år

Erfaring: 32 år

◆ Biografi

- ◆ Mariann er en streng, men hyggelig og varm person. Etter å ha fullført sin utdanning innen økonomi og administrasjon, kom hun raskt ut i jobb og har jobbet i flere ulike bedrifter som regnskapsfører. Hun trives godt arbeidet.

◆ Holdning

- ◆ Liker seg godt inne i varmen, og trives best på sofaen foran TVen.
- ◆ Liker å finne ut av feil selv, og lære nye ting mest mulig selvstendig før hun spør om hjelp.

◆ Mål

- ◆ Ønsker å få tatt i bruk verktøy som hjelper hun med regnskap.
- ◆ Har lyst å bli bedre kjent med ny teknologi som AI. Hun skjønner ikke helt hvordan det fungerer, men ønsker å bli bedre.

◆ Erfaring

- ◆ Har hørt mye nytt om AI i nyhetene, og tenker det hadde vært perfekt å ta i bruk i jobben om hun hadde fått mulighet til det.

**Elisabeth Grønnskog**

Alder: 26 år - f. 1998

Utdanning: Tjeneste design

Rolle: UX design

Ansiennitet i bedriften: 3 år

Erfaring: 3 år

◆ Biografi

- ◆ Elisabeth er en kreativ og lidenskapelig UX-designer. Etter å ha fullført sin utdanning innen tjenstedesign, og med 3 års erfaring i konsultantselskap, har hun etablert seg som en talentfull profesjonell innen brukeropplevelsesdesign.

◆ Holdning

- ◆ Aktiv og energisk person som ikke liker å sitte stille.
- ◆ Liker å ta i bruk mennesker rundt seg for å finne hjelp til ting hun lurer på.

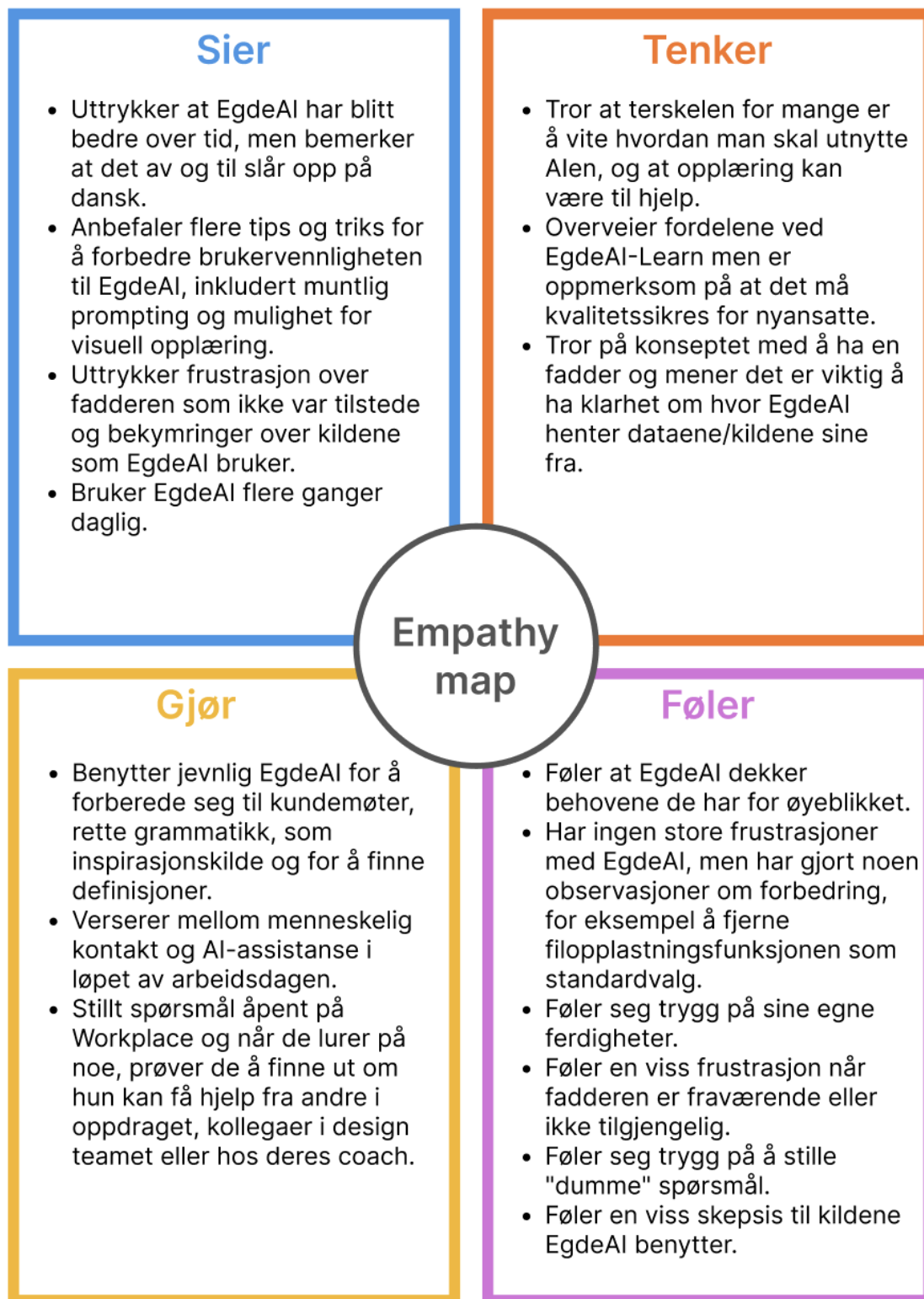
◆ Mål

- ◆ Ønsker å forbedre brukeropplevelser gjennom innovative og brukervennlige designløsninger.
- ◆ Kontinuerlig utvikle sine ferdigheter og holde seg oppdatert på nye designtrender og teknologier.

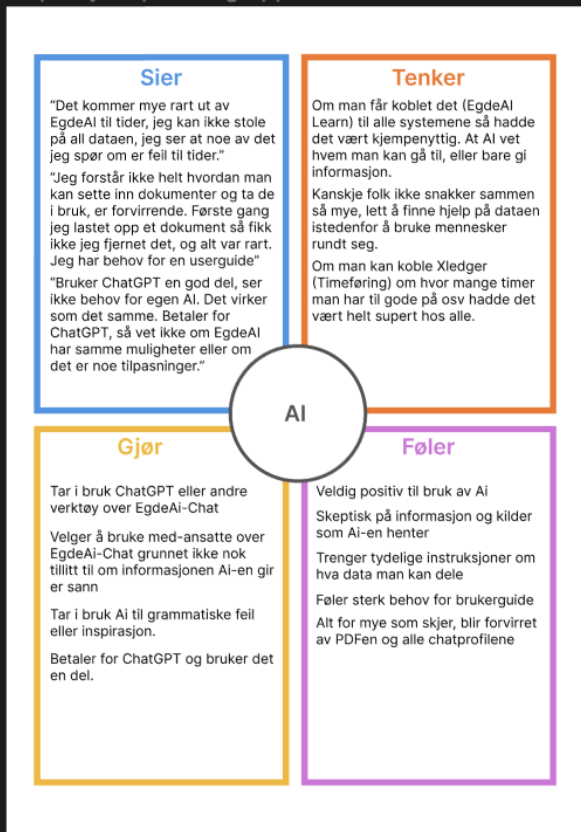
◆ Erfaring

- ◆ Har hørt at AI-verktøy kan bidra til mer effektiv arbeid, men har valgt å ikke ta det i bruk i eget arbeid grunnet manglende tillit til AI.

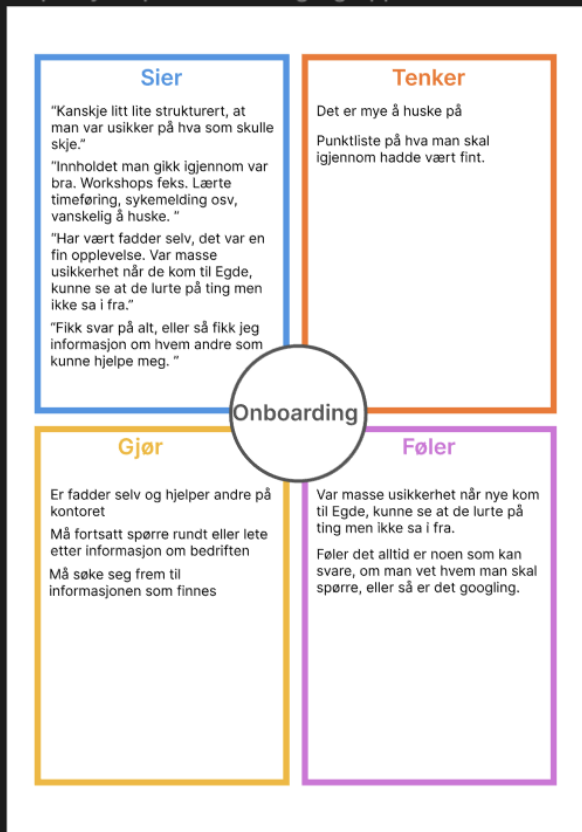
Vedlegg 14 - Empatikart



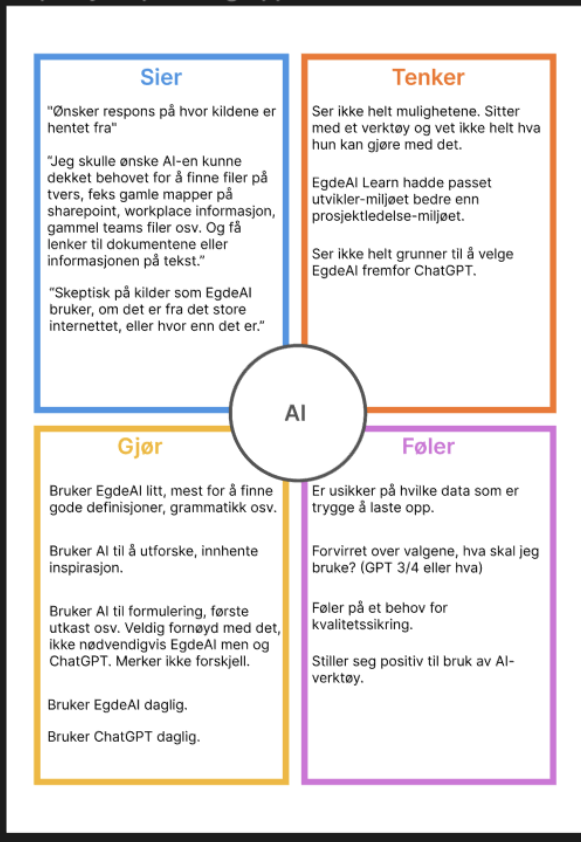
Empathy map - AI - gruppe 2



Empathy map - Onboarding - gruppe 2



Empathy map - AI - gruppe 1



Empathy map - Onboarding - gruppe 1



Vedlegg 15 - Brukerhistorier

As a [type of user],
I want [an action]
so that [a benefit/a value].

som en nyansatt,
ønsker jeg hjelp til å lære ulike verktøy
så jeg kan bidra i prosjektteamet mitt.

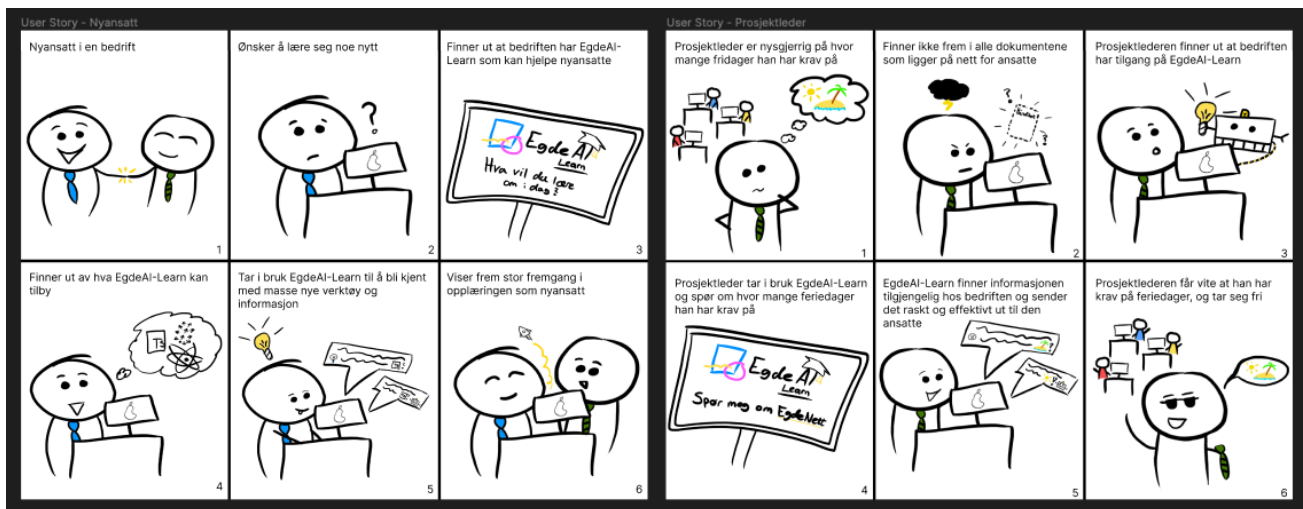
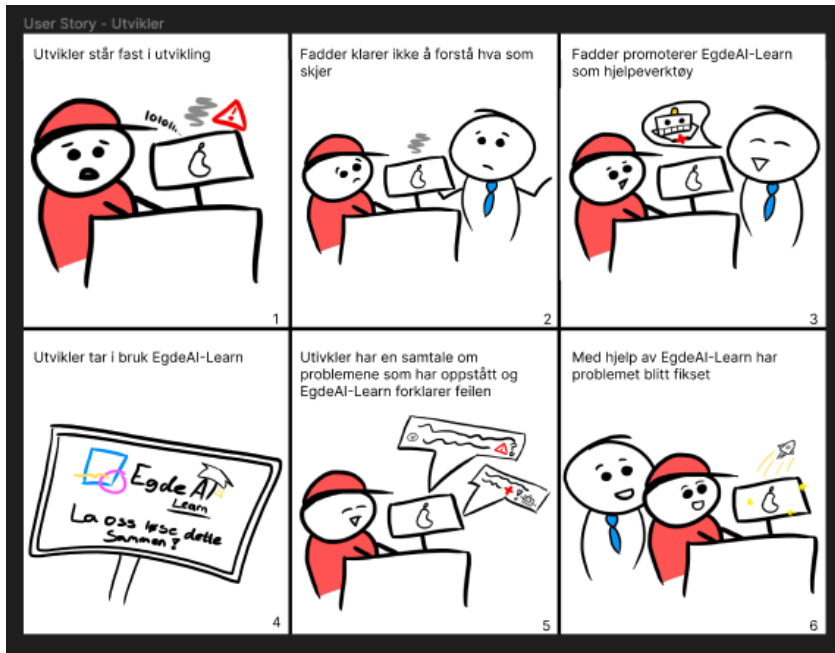
som en prosjektleder,
ønsker jeg lett tilgang på informasjon i EgdeNett
så jeg kan spare verdifull tid.

som en ux-designer,
ønsker jeg å få inspirasjon og øke kreativitet
så jeg kan komme raskt i gang med arbeid.

som en utvikler,
ønsker jeg hjelp med kode
så jeg slipper å bruke tid på bug-fixing.

som en regnskapsfører,
ønsker jeg å hente informasjon om bedriften
så jeg ikke trenger å søke opp manuelt.

Vedlegg 16 - Visuell brukerhistorie



Vedlegg 17 - Factor-modell

FACTOR-modell	
Funksjonalitet (Functionality)	Digital tjeneste for effektivisering av innhenting og prosessering av informasjon fra digitale ressurser ved hjelp av AI-teknologi.
Applikasjonsdomene (Application Domain)	Optimalisere nyansettelsesprosesser, onboarding og generell læring for Egdes ulike prosjektteams. Effektivt kunne innhente informasjon fra EgdeNett. Potensiale for å brukes av andre bedrifter.
Forhold (Conditions)	<p>For at tjenesten skal kunne tilby gode svar basert på brukerens inndata er det viktig at dataene og ressursene i kunnskapsbasen er godt strukturert og enkelt å behandle.</p> <p>Ved bruk:</p> <p>En stødig internettforbindelse kreves for å bruke tjenesten. Tjenesten fungerer også mer optimalt basert på kvaliteten på brukerens inndata.</p>
Teknologi (Technology)	<p>Som en web-basert tjeneste bruker applikasjonen flere web-utviklings teknologier.</p> <p>Rammeverk: Next.js, Vercel, React</p> <p>Frontend: Tailwind CSS</p> <p>Backend: TypeScript, OpenAI API</p> <p>Database: PineCone VectorDB, AzureAI Search</p>
Objekt (Object)	<p>Det er to primære objekter i systemet.</p> <p>En ChatGPT-modell og en kunnskapsbase. GPT-modellen forvalter informasjon til brukeren basert på inndata.</p> <p>Informasjon som blir forvaltet kommer fra kunnskapsbasen som holder på dataene og ressursene.</p>

Vedlegg 18 - MoSCoW-analyse (original og oppdatert)

EgdeAI-Learn | MoSCoW analyse

Must have	Should have	Could have	Will not have
Samtale/chat funksjon — Automatisert produksjon av læringsoppgaver — Bruker autorisering og autentisering — Kunnskapsbase — GPT-minne	Team-profiler — Prompt profiling — Chathistorikk — Adminopplastning av filer — Utklippstavle-operasjoner	Bildegenerering for å støtte læringsoppgaver — Egendefinert designtilpassning for team/kunder — Flere tjenestenivåer — AI-profiler — Stilligsbasert informasjonstilgjengelighet	Brukeropplastning av filer — GPT-modell opptrening

EgdeAI-Learn | MoSCoW analyse

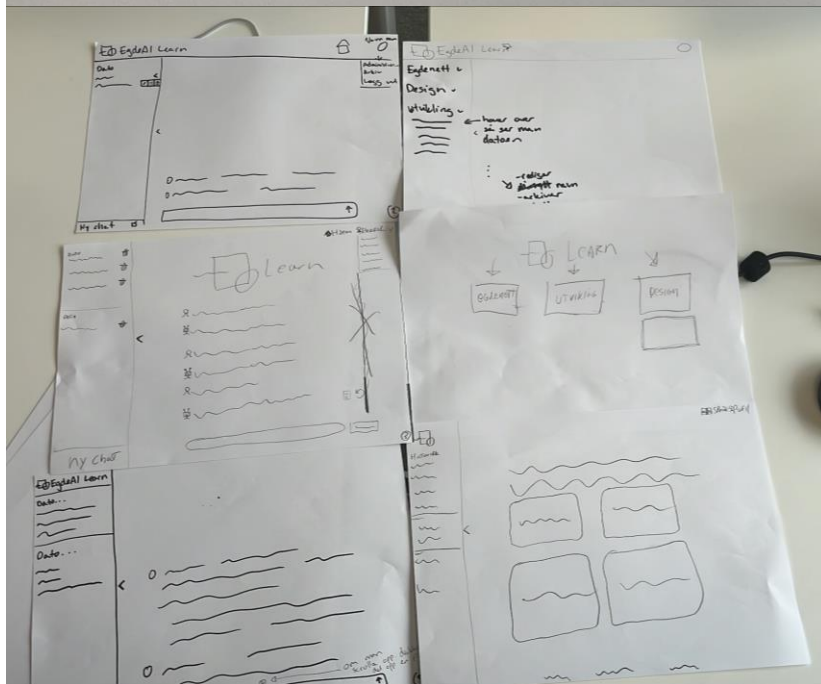
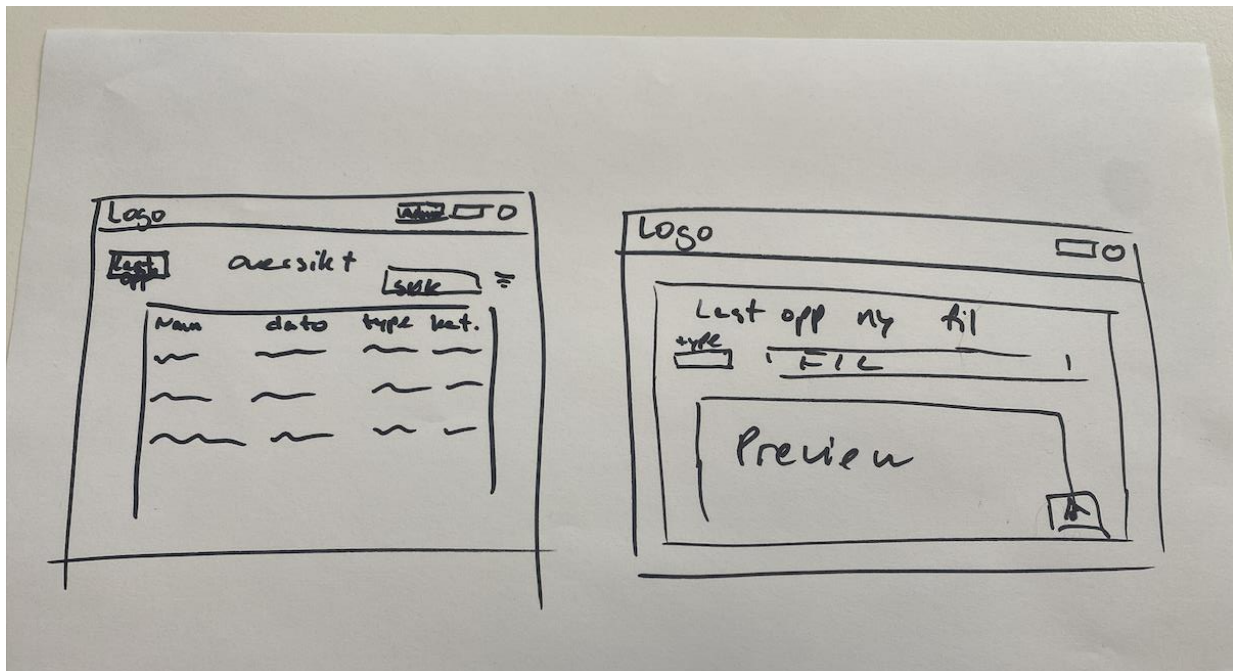
Must have	Should have	Could have	Will not have
Samtale/chat funksjon — Bruker autorisering og autentisering ↓ Systemroller — GPT-minne — Standardisert AI prompt profil — Kunnskapsbase ↓ Separering av kunnskapsbaser	Samtale-historikk — Adminopplastning av kunnskapsdata ↓ Administrator kontrollpanel — Automatisert produksjon av læringsoppgaver — Markdown format på samtale tekst — Forhåndsvisning av opplastede filer — Omvisningstur — Tilpassning av brukergrensesnitt — Prompt-wizard	Bildegenerering for å støtte læringsoppgaver — Egendefinert designtilpassning for team/kunder — Åpen produktdemo for mulige kunder ↓ Flere produktnivåer — Rollebasert informasjonstilgjengelighet — Egendefinert Team-profiler — Utklippstavle-operasjoner — Arkiv for slettede filer — Brukerguide	Ikke-autorisert opplastning av filer — GPT-modell opptrening på brukerdata — Opptrening av egen GPT-modell — Arkivering av samtale-historikk — Personlig kunnskapsbase — AI-modell profil

Vedlegg 19 - Systemkriterier

- *Brukervennlig* dreier seg om hvor enkelt systemet er å ta i bruk og hvor intuitivt det føles for brukerne. Hvor enkelt og effektivt produktet kan tas i bruk av kunden er avgjørende for dets suksess.
- *Sikker* er et kritisk kriterium som handler om sikkerheten rundt systemet, altså hvor beskyttet systemet og dets data er mot potensielle trusler.
- *Korrekt* handler om hvorvidt tjenesten oppfyller kravene satt av produkteier.
- *Troverdig* går ut på at funksjonene i systemet utfører handlingen som er etterspurt av produkteier.
- *Forståelig* dreier seg om mengden innsats som kreves for å forstå systemet.
- *Effektiv* fokuserer på optimal utnyttelse av den tekniske plattformens potensiale.
- *Vedlikeholdbar* handler om hvor krevende det vil være å lokalisere og fikse problemer og feil i systemet.
- *Fleksibel* dreier seg om hvor ressurskrevende det vil være å gjøre endringer i systemet i etterkant av utviklingen.
- *Gjenbrukbar* fokuserer på potensialet for gjenbruk av deler av systemet i andre relaterte system.
- *Testbar* handler om muligheten til å teste systemet, hvor ressurskrevende det er å sjekke og forsikre at systemet fungerer slik det var tiltenkt.
- *Flyttbar* dreier seg om hvor ressurskrevende det vil være å flytte systemet til en annen teknisk plattform.
- *Samvirkosom* handler om hvor ressurskrevende det vil være å koble systemet opp mot andre systemer.

(Mathiassen, 2018)

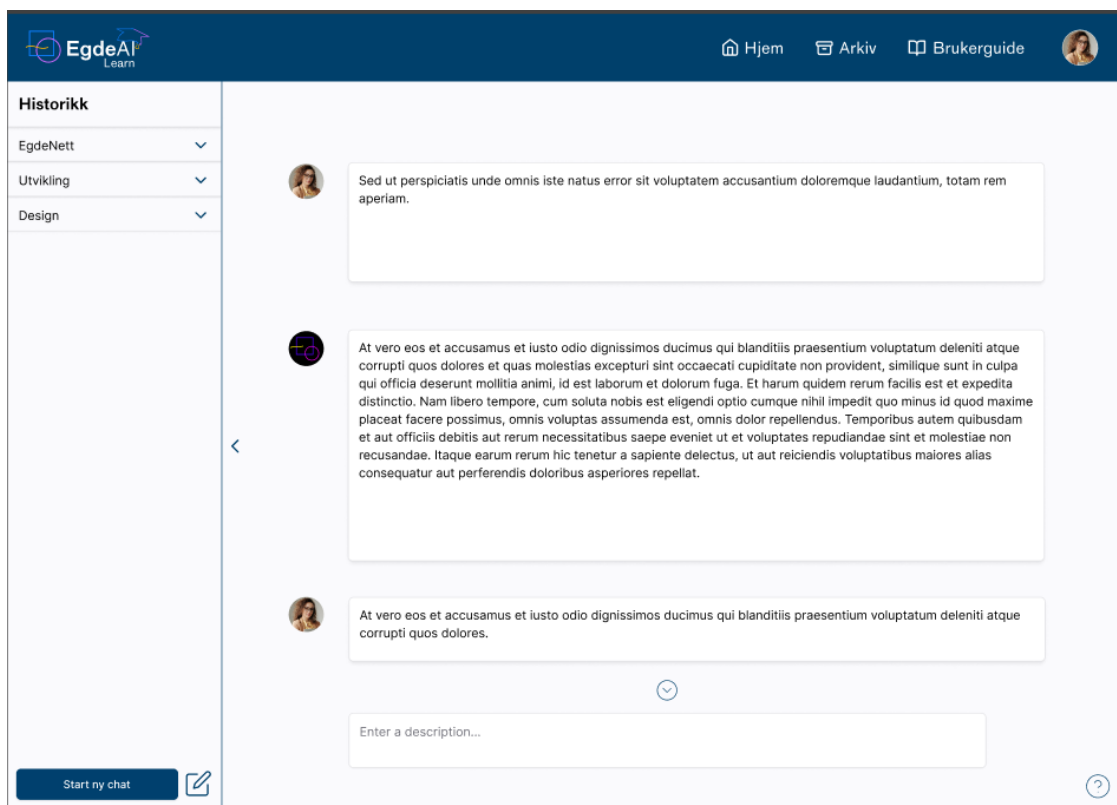
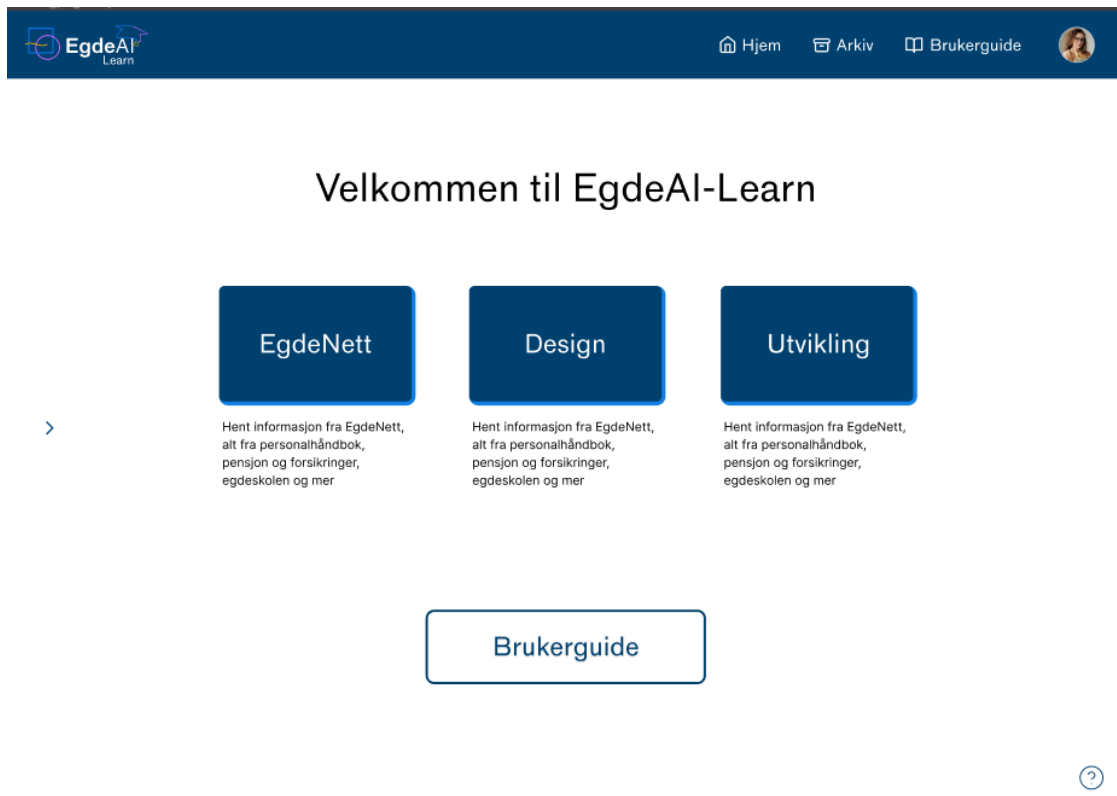
Vedlegg 20 - Sketches





Vedlegg 21 – Lo-Fi Wireframes




Vedlegg 22 – Hi-Fi Wireframes






[Hjem](#) [Arkiv](#) [Brukerguide](#) 


Brukerguide




Hvorfor ta i bruk EgdeAi-Learn




Hvordan å snakke med Ai




Template text




Template text





Template text













Template text






[Hjem](#) [Arkiv](#) [Brukerguide](#) 

Arkiv

Dato	Tittel	
29.januar 2024	Dignissimos ducimus dignissimos ducimus	 
29.januar 2024	Dignissimos ducimus dignissimos ducimus ducimus	 
27.januar 2024	Dignissimos ducimus dignissimos	 
> 21.januar 2024	Dignissimos ducimus dignissimos ducimus	 
21.januar 2024	Dignissimos ducimus	 



EgdeAI Learn

[Hjem](#)
[Arkiv](#)
[Brukerguide](#)

Hvorfor ta i bruk EgdeAI-Learn



Hvorfor ta i bruk EgdeAI-Learn

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil inpedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Hvorfor ta i bruk EgdeAI-Learn

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil inpedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.





Hvorfor ta i bruk EgdeAI-Learn

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil inpedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

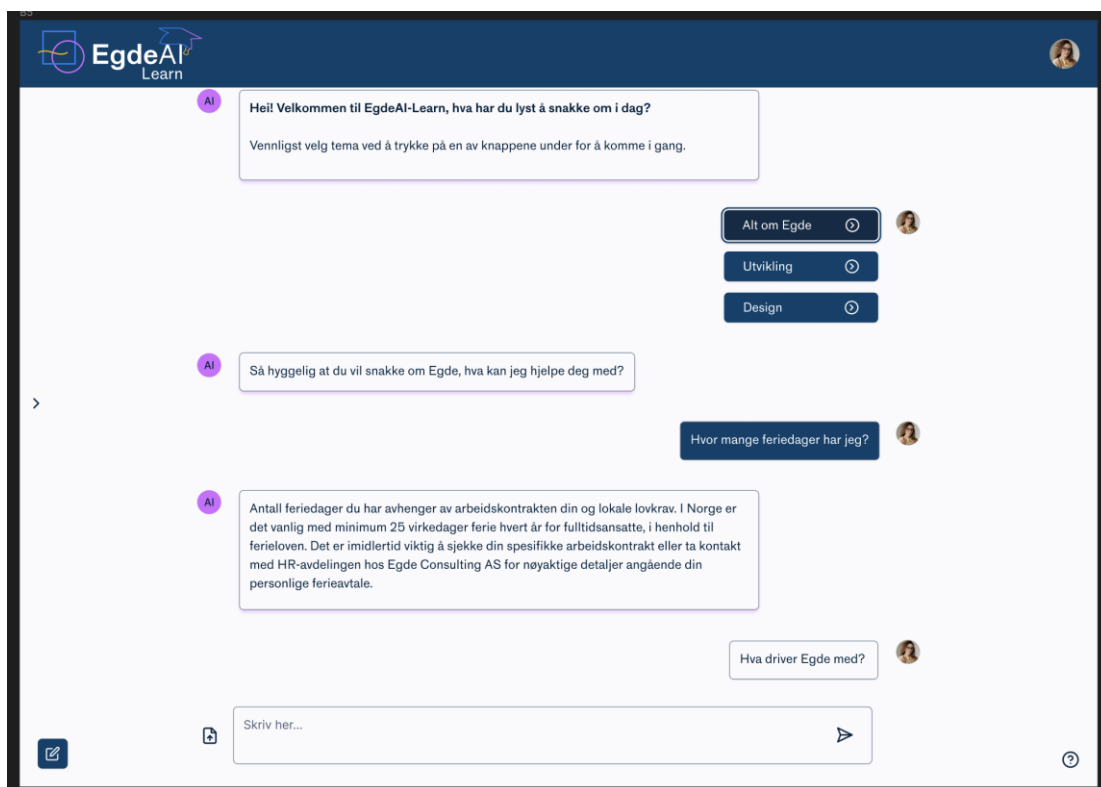
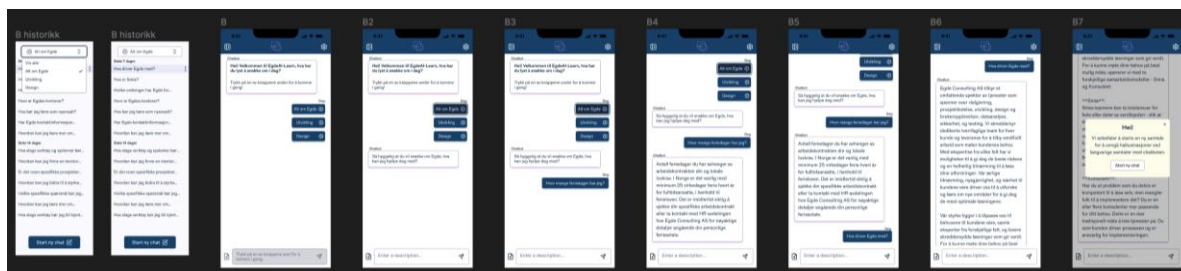
Hvorfor ta i bruk EgdeAI-Learn

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil inpedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.





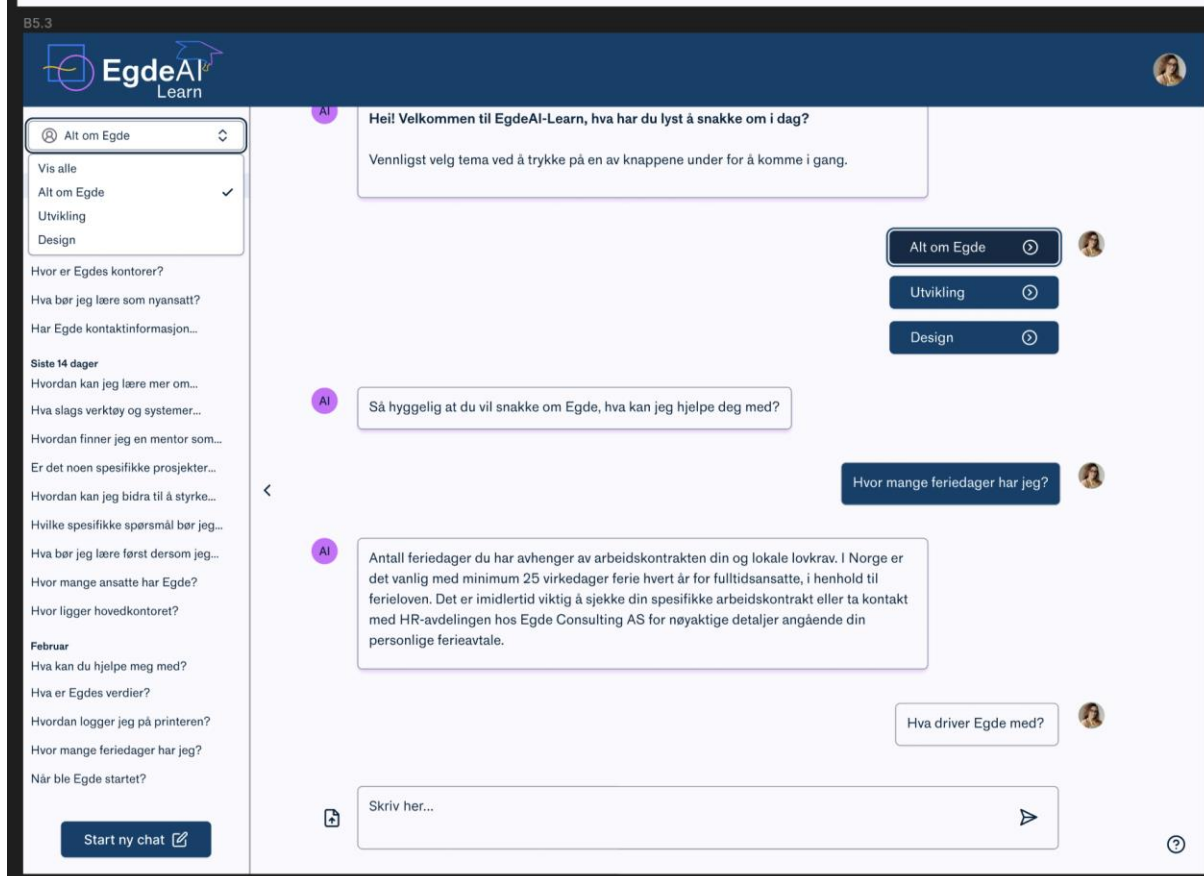
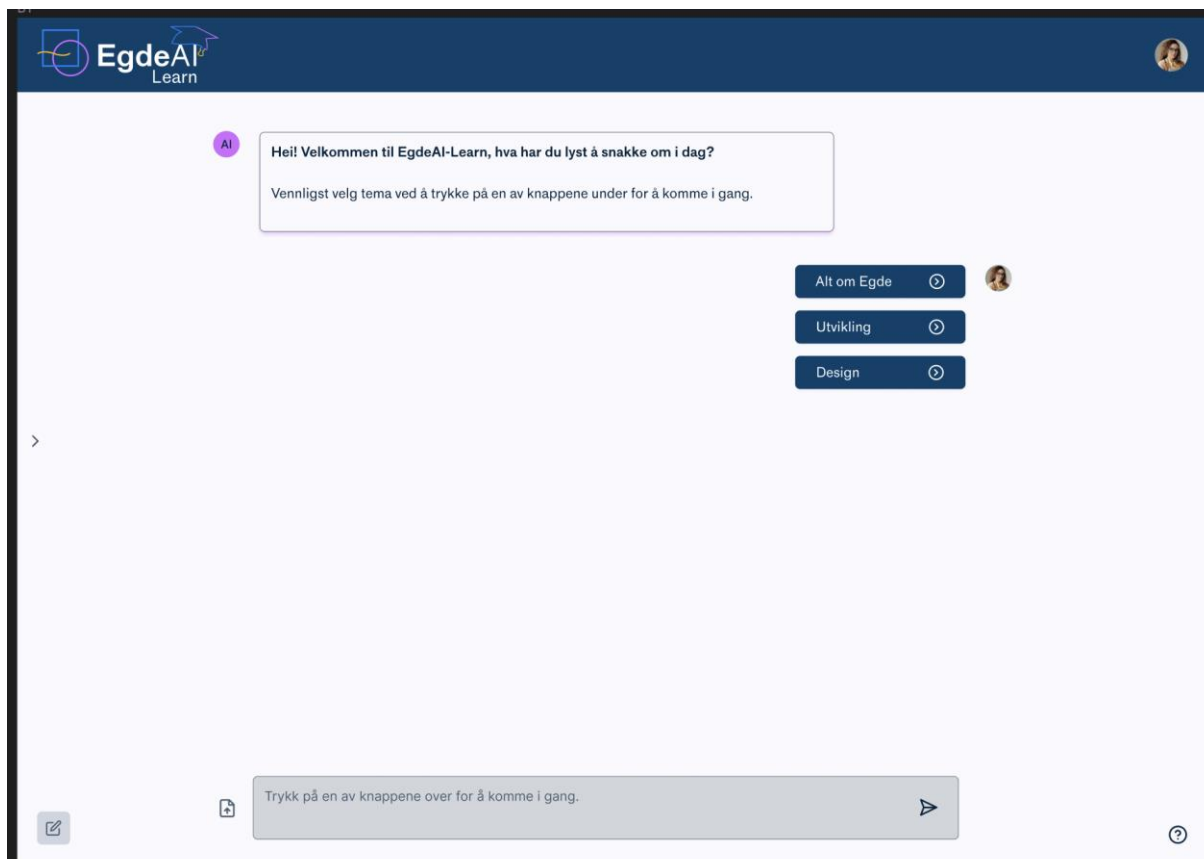
Vedlegg 23 - Prototype Chat



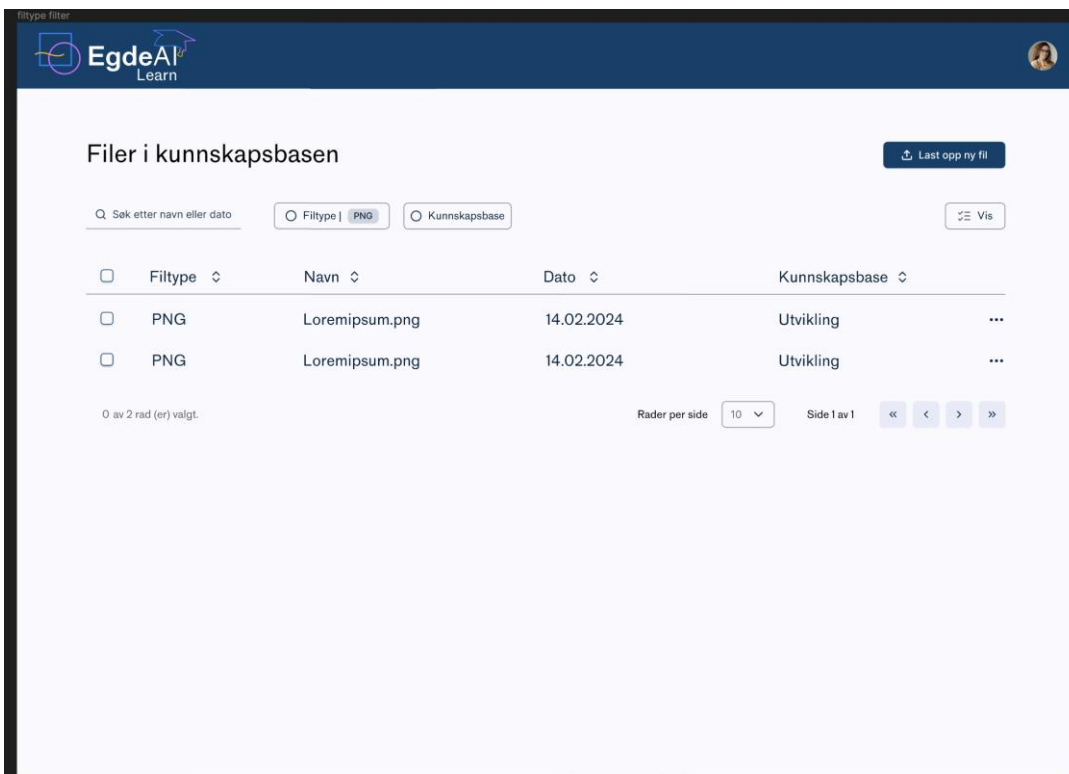
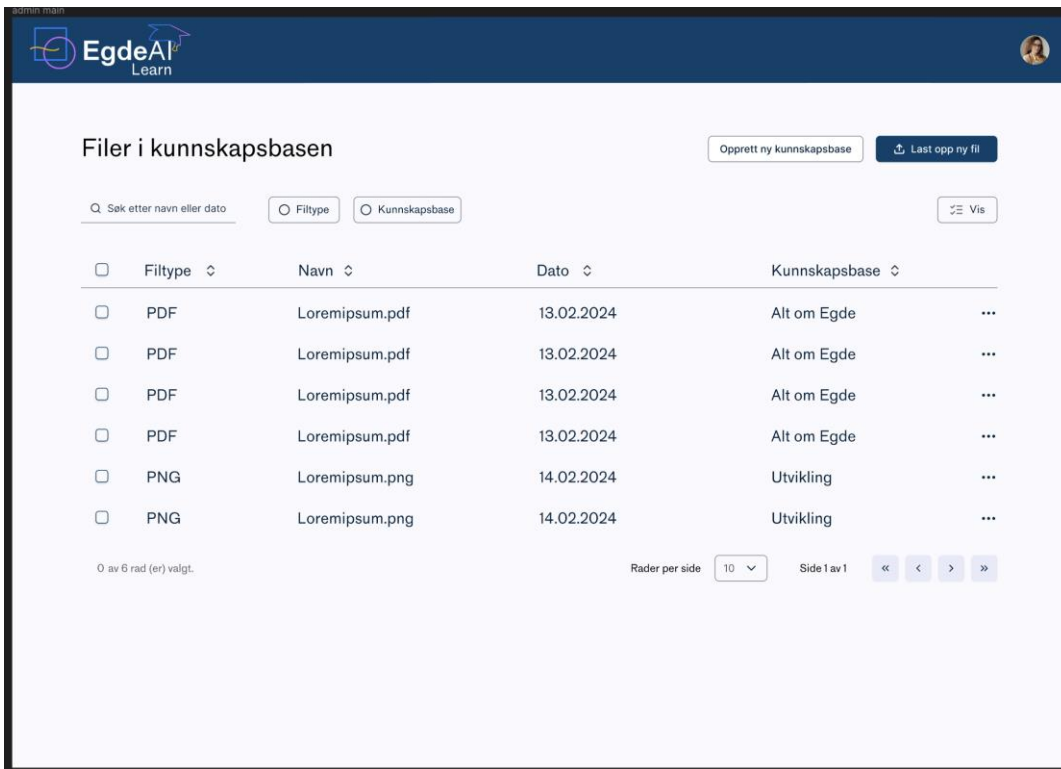
Link til Figma-prototype

Desktop - <https://www.figma.com/proto/Gew52OP1cnI28nQOhmEwiL/Prototype?page-id=59%3A3010&node-id=692-18063&viewport=-1771%2C219%2C0.06&t=lvPgbxNP3NPQiCiD-1&scaling=min-zoom&starting-point-node-id=692%3A18063&show-prot-sidebar=1>

Mobil - <https://www.figma.com/proto/Gew52OP1cnI28nQOhmEwiL/Prototype?page-id=28%3A6267&node-id=240-5903&viewport=4535%2C-4451%2C0.76&t=2Sn61PVt9kl2IF1Z-1&scaling=min-zoom&starting-point-node-id=240%3A5903&show-prot-sidebar=1>



Vedlegg 24 - Prototype admin page



Filer i kunnskapsbasen

📄 Last opp ny fil

🔍 Søk etter navn eller dato

📄 Filtype 📄 Kunnskapsbase

☑️ Vis

<input type="checkbox"/>	Navn	Dato	Kunnskapsbase	
<input type="checkbox"/>	Loremipsum.pdf	13.02.2024	Alt om Egde	⋮
<input type="checkbox"/>	Loremipsum.pdf	13.02.2024	Alt om Egde	⋮
<input type="checkbox"/>	Loremipsum.pdf	13.02.2024	Alt om Egde	⋮
<input type="checkbox"/>	Loremipsum.pdf	13.02.2024	Alt om Egde	⋮
<input type="checkbox"/>	Loremipsum.pdf	14.02.2024	Utvikling	⋮
<input type="checkbox"/>	Loremipsum.pdf	14.02.2024	Utvikling	⋮

0 av 6 rad (er) valgt.

Rader per side: 10 Side 1 av 1

Filer i kunnskapsbasen

📄 Last opp ny fil

🔍 Søk etter navn eller dato

📄 Filtype

📄 PDF

📄 PDF

📄 PDF

📄 PDF

📄 PNG

📄 PNG

0 av 6 rad (er) valgt.

Loremipsum ✕

[preview]

Kunnskapsbase: Tittel

Filtype: PDF

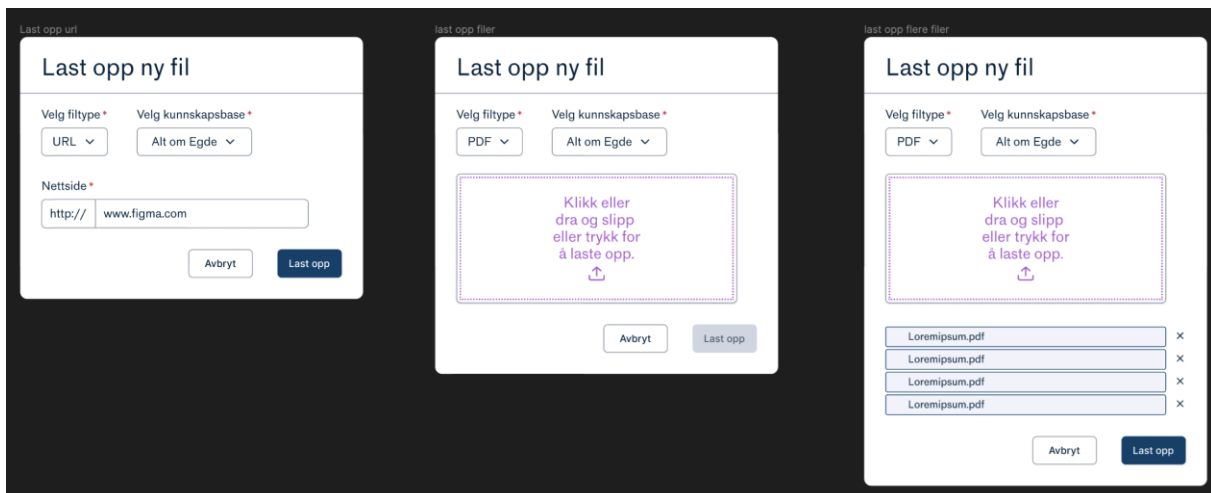
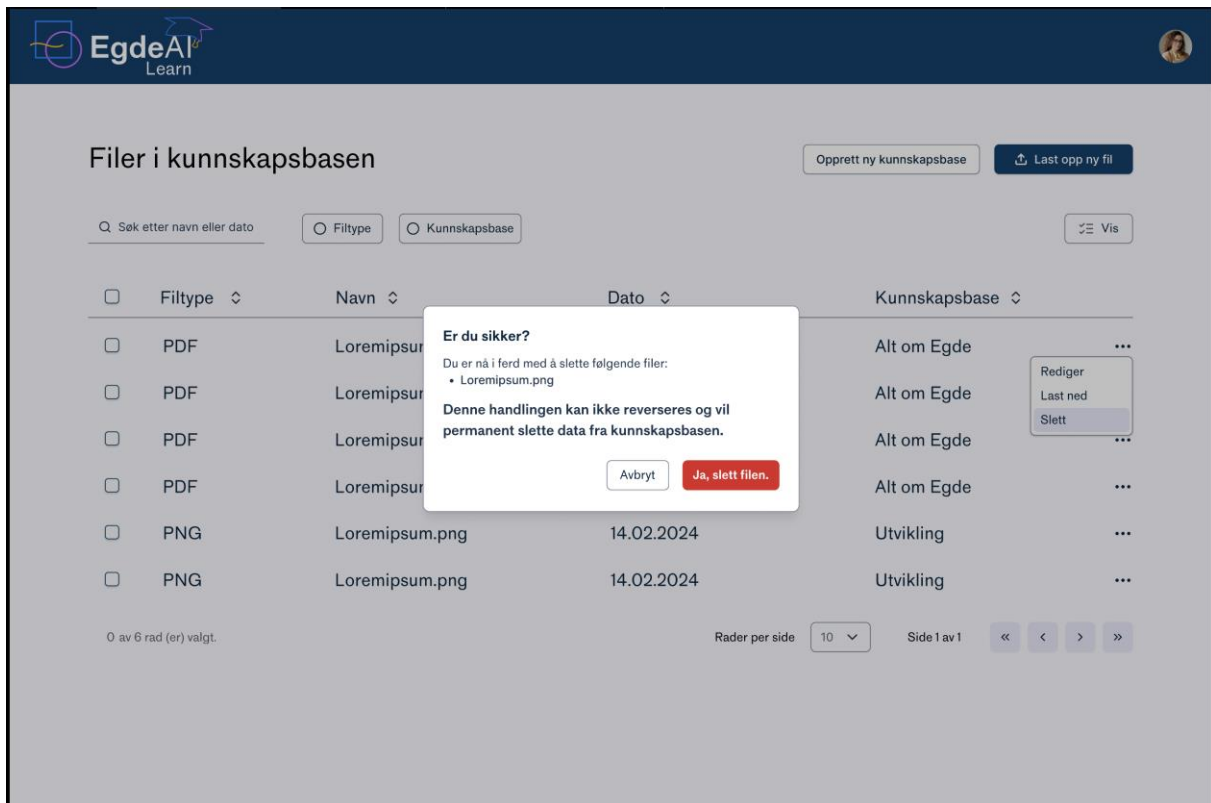
Dato: 04.03.24

Lastet opp av: Anonym

Kommentar:

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium.

Rader per side: 10 Side 1 av 1



Link til Figma-prototype

Administratorside - <https://www.figma.com/proto/Gew52OP1cnI28nQOhmEwiL/Prototype?page-id=170%3A19103&node-id=245-16844&viewport=-77%2C111%2C0.14&t=NYUY3j5ZwhmG2xSZ-1&scaling=min-zoom&starting-point-node-id=245%3A16844>

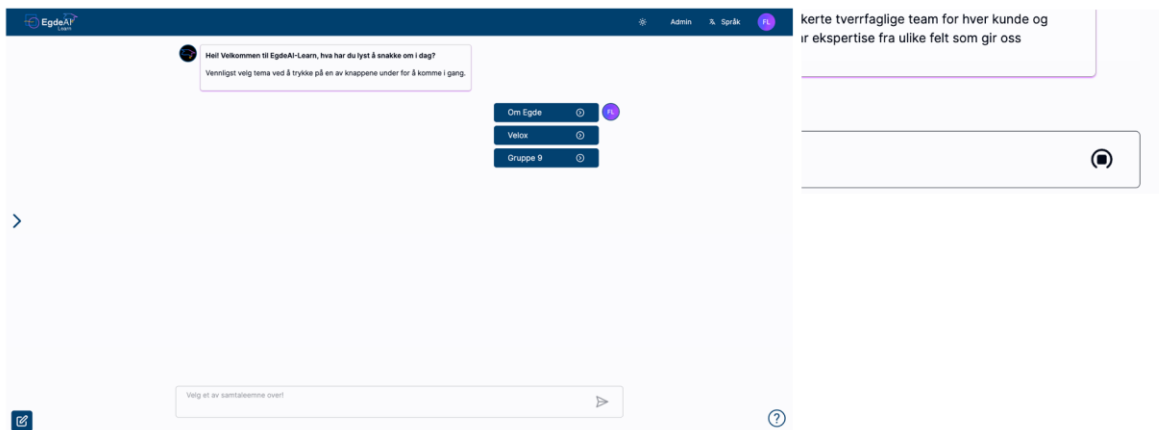
Vedlegg 25 - Definisjon av Benyons designprinsipper

1. Visibility – gjør det synlig for brukeren hvilke funksjoner som er tilgjengelig og hva systemet gjør for øyeblikket.
2. Consistency – konsekvent bruk av designelementer, bruk av elementer som brukerne kjenner igjen.
3. Familiarity – bruk av språk og symboler brukerne kjenner igjen.
4. Affordance – gjør det synlig hva noe er, og hva det gjør.
5. Navigation – gjør det enkelt og intuitivt for brukeren å navigere seg gjennom applikasjonen.
6. Control – la brukeren føle en form for kontroll.
7. Feedback – gi tilbakemelding om hva som skjer og hvilken påvirkning handlingene deres har.
8. Recovery – gi brukerne mulighet til å angre handlinger.
9. Constraints – begrense og forhindre brukerfeil.
10. Flexibility – gi brukerne flere muligheter til å utføre en handling.
11. Style – Tilfredsstillende design, passende farger og pent oppsett.
12. Conviviality – systemet bør oppleves som vennlig og høflig.

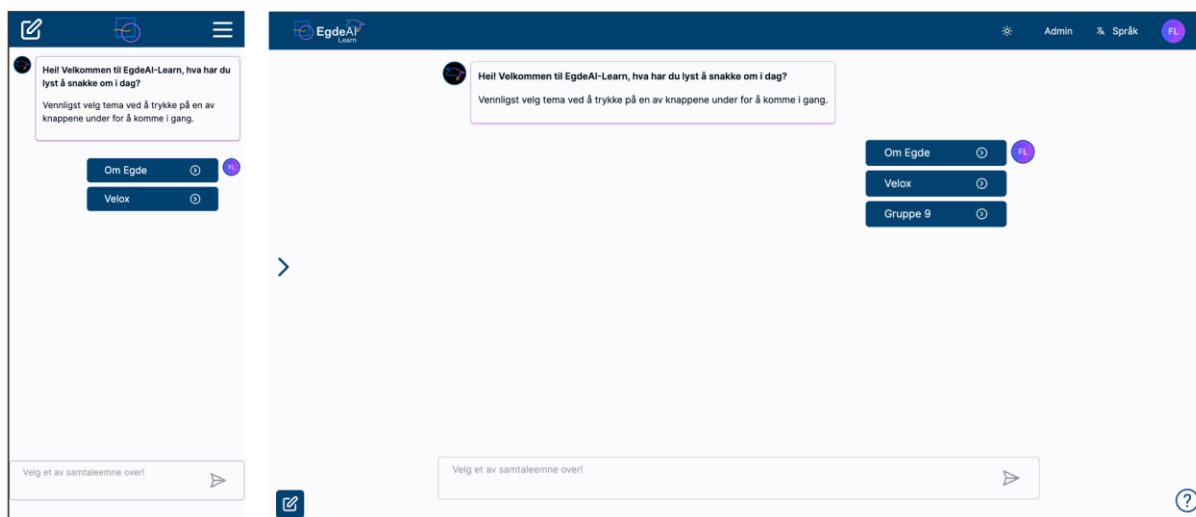
(Mathiassen, 2018)

Vedlegg 26 - Eksempler på bruk av Benyons designprinsipper

Visibility – Brukerne kan for eksempel se ikonet på venstre side for å åpne sidepanelet, og knappen nederst for å starte ny samtale. Oppe i høyre hjørne kan brukerne velge å endre lysmodus og språk. Når brukeren har en samtale med AI-en, vil et ikon vises for å gjøre det tydelig for brukeren at AI-en «tenker» (figur under). Inni dette ikonet vil det også være en stopp knapp, så brukeren kan avbryte genereringen.

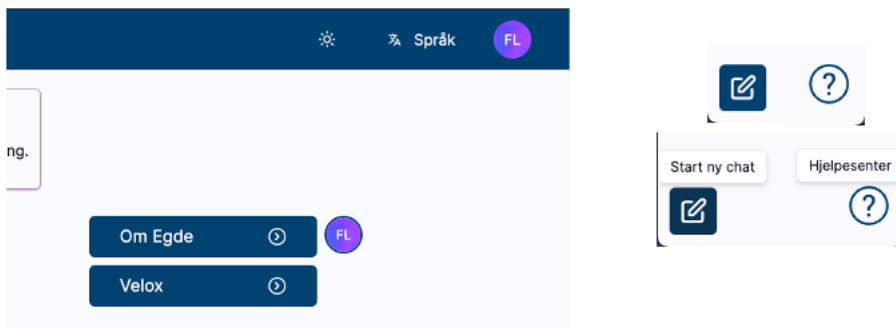


Consistency – For at designet skal være brukervennlig og enkelt å forstå var det viktig for gruppen å være konsistent i bruk av farger, skrifttyper og designelementer. Dette bidrar til å unngå at designet føles forstyrrende eller forvirrende ut. Brukeren slipper å stadig måtte lære nye måter å gjøre ting eller utføre oppgaver; når de først er kjent med systemet kan de bruke det uten problemer (UX Glossary, u.å.). Figurene under viser at bruker blir møtt med like designelementer på desktop og mobil.



Familiarity – Bruker blir møtt med enkle og forståelige ikoner. Bruker vet at spørsmålstegnet indikerer at om de trenger hjelp kan de trykke der. I tillegg til at ikonene er kjente for brukerne, vil

det også vises en beskrivende tekst når man hover over ikoner og knapper som ikke inneholder tekst.

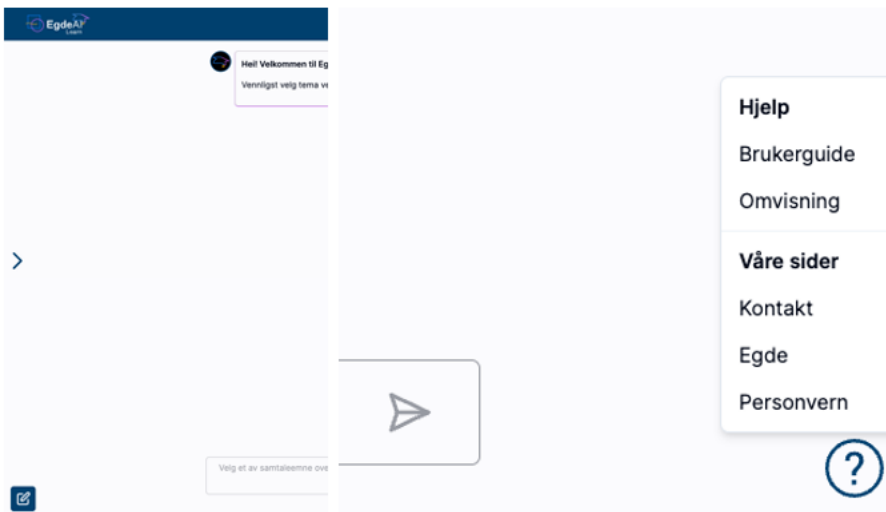


Affordance – Refererer til det å kommunisere til brukerne funksjonaliteten eller hensikten bak et element (Uxcel, u.å.). Dette kriteriet kan illustreres ved å se på første interaksjonen brukerne vil ha i chat. Når de først kommer inn blir brukeren informert om at de må velge tema ved å trykke på en av knappene under for å komme i gang. Knappene er tydelige og kommuniserer til brukerne at de kan trykkes på.

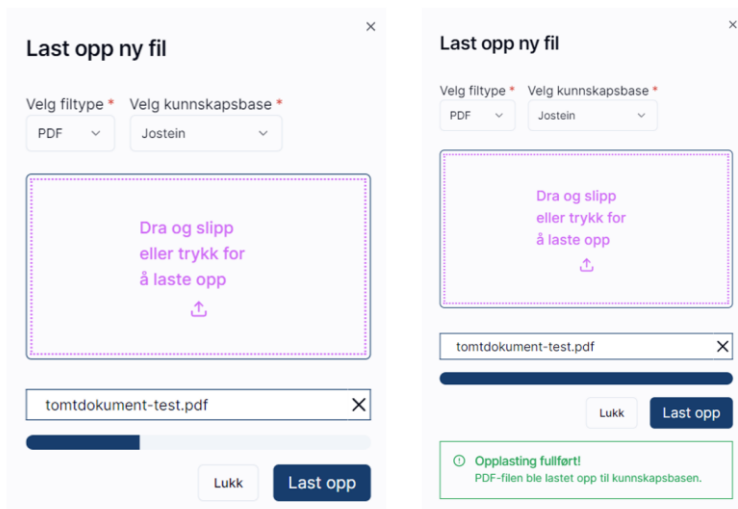


Navigation – Handler om hvordan brukerne navigerer seg gjennom systemet. Det skal være klare og tydelige indikasjoner på hvor brukeren må trykke for å komme seg dit de vil. Et eksempel på dette er hjelpesenter-knappen nede i høyre hjørne. Her vil brukerne finne hjelpesider som brukerguide og mulighet for å få en omvisning av systemet. Et annet eksempel er pil-ikonet på venstre side som viser til at de kan klikke på den for å åpne et sidepanel. Når bruker holder

musetasten over får de opp en tekst som sier «historikk». Nederst i venstre hjørne finner brukerne også knappen for å starte en ny samtale.



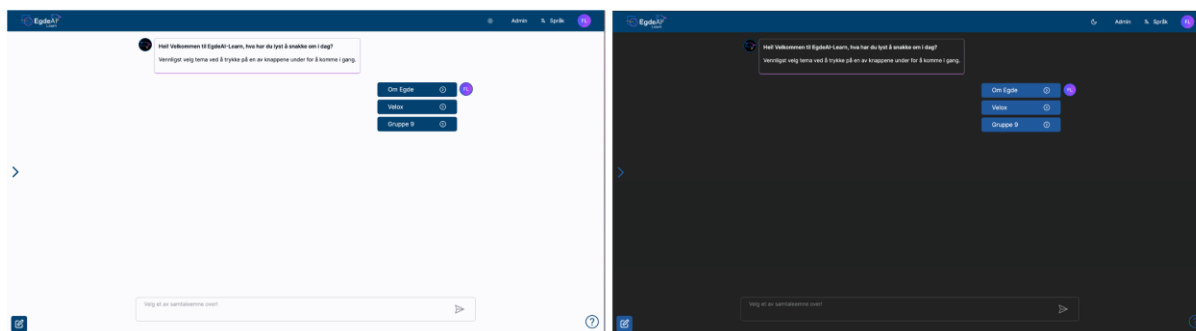
Feedback – Dette prinsippet er spesielt relevant for administrasjonssiden. Når brukerne skal laste opp filer i en kunnskapsbase, vil de få tilbakemelding om noe står i veien for at filen skal bli lastet opp. Dette vil skje for eksempel viss bruker glemmer å velge hvilken kunnskapsbase filen skal tilhøre. Brukerne vil også få tilbakemelding når filen har blitt vellykket lastet opp.



Constraints – Dette kriteriet handler om å begrense brukerfeil og hindre upassende handlinger. Et eksempel på hvordan dette har blitt tatt hensyn til er input-feltet i chat før brukeren har valgt kunnskapsbase. Bruker vil ikke kunne sende meldinger til chatboten før de har valgt samtaleemne. Bruker får også beskjed om hva som må gjøres før de kan begynne å skrive inn i feltet.

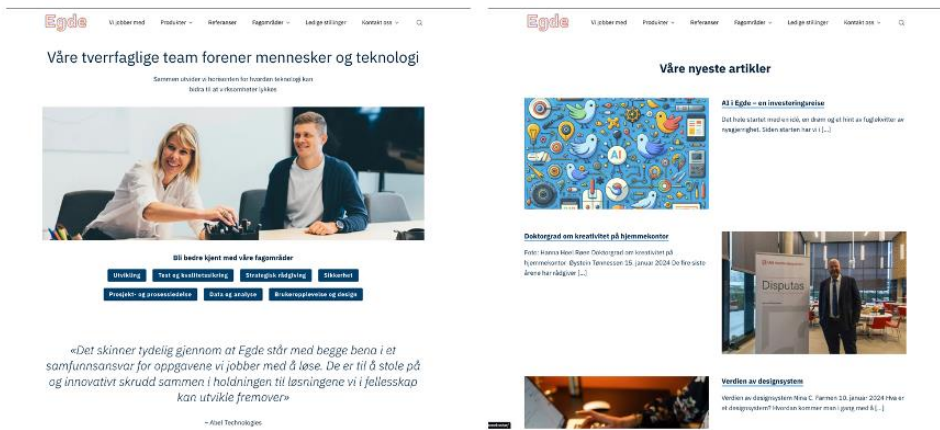


Flexibility – Bruker har mulighet til å velge lys eller mørk modus, se figurene under. Bruker har også mulighet til å velge språk (norsk/engelsk). Andre eksempler på fleksibilitet kan ses i administrasjonssiden. Bruker har flere valgmuligheter til å sortere filene. Andre valgmuligheter brukerne får for å gjøre opplevelsen bedre er i chathistorikken. Bruker får muligheten til å enkelt sortere samtaler etter hvilken kunnskapsbase de tilhører.



Style – Refererer til det estetiske ved designet. Produkteier kom med ønske om fargepalett og generell stil, og viste til Egdes nettside som en inspirasjonskilde. Systemet tar i bruk Egdes farger som vist under.





Conviviality – Dette kriteriet handler om at systemet skal oppleves som høflig og vennlig. Tekst og symboler skal ikke oppleves som nedlatende eller angripende ovenfor brukerne. Eksempler fra systemet kan være meldingene fra chatboten (figuren under), og tilbakemeldingene brukerne får når de laster opp filer i administrasjonssiden.



Vedlegg 27 - WCAG rapport

EgdeAI-Learn

About the Evaluation

Report Creator

Learn - ux team

Evaluation Commissioner

Learn - dev team

Evaluation date

Tue Mar 26 2024

Executive Summary

Not provided

Scope of the Evaluation

Website name

EgdeAI-Learn

Scope of the website

All web content of desktop and mobile version of the website located at <https://learn.ai.egde.no/no> and <https://learn.ai.egde.no/en> on light mode. Does not include pages only available for administrators.

WCAG Version

2.2

Conformance target

AA

Accessibility support baseline

Not provided

Additional evaluation requirements

The report will include a description of the problems and observations on each criterion. The evaluation will cover all web pages and web content of the web site.

Detailed Audit Results

Summary

Reported on 55 of 55 WCAG 2.2 AA Success Criteria.

- 38 Passed
- 2 Failed
- 0 Cannot tell
- 15 Not present
- 0 Not checked

All Results

1 Perceivable

1.1 Text Alternatives

Success Criterion	Result	Observations
1.1.1: Non-text Content	Result: Passed	Observations: Alle ikon/knapper uten tekst har tooltip tekst, så når bruker hoverer over vil de få en beskrivelse.

1.2 Time-based Media

Success Criterion	Result	Observations
1.2.1: Audio-only and Video-only (Prerecorded)	Result: Not present	Observations: ingen forhåndsinnspilt video/audio innhold.
1.2.2: Captions (Prerecorded)	Result: Not present	Observations: ingen forhåndsinnspilt video/audio innhold.
1.2.3: Audio Description or Media Alternative (Prerecorded)	Result: Not present	Observations: ingen forhåndsinnspilt video/audio innhold.
1.2.4: Captions (Live)	Result: Not present	Observations: ingen forhåndsinnspilt video/audio innhold.
1.2.5: Audio Description (Prerecorded)	Result: Not present	Observations: ingen forhåndsinnspilt video/audio innhold.

1.3 Adaptable

Success Criterion	Result	Observations
1.3.1: Info and Relationships	Result: Not present	Observations: Bruker siden inneholder ikke forms hvor brukeren må velge input. Valg av * for å vise hva som er påbudt, farger av forms, eller annet er ikke relevant.
1.3.2: Meaningful Sequence	Result: Passed	Observations: På siden er det lagt opp til at navigasjonsbare ligger øverst, menyer på siden, og hoved innhold på midten for brukeren. Alle elementene er posisjonert basert på familiarity. Dette gjør at innholdet på siden blir mer forstått av flere folk.
1.3.3: Sensory Characteristics	Result: Passed	Observations: Knappene er enkle for brukeren å se og forstå.
1.3.4: Orientation	Result: Failed	Observations: Delvis funksjonell på landskap på mobil. (chaten er mulig å bruke, men sidepanelene er ikke fullt tilgjengelige)
1.3.5: Identify Input Purpose	Result: Passed	Observations: forhåndsdefinerte input valg tilgjengelige.

1.4 Distinguishable

Success Criterion	Result	Observations
1.4.1: Use of Color	Result: Passed	Observations: Fargene er ikke brukt alene for å formidle informasjon, indikere en handling,

		fremkalle et svar eller skille et visuelt element.
1.4.2: Audio Control	Result: Not present	Observations: Ingen lyd spilles av automatisk.
1.4.3: Contrast (Minimum)	Result: Passed	Observations: Kontrast tekst: 15,39:1
1.4.4: Resize text	Result: Passed	Observations: Tekst kan forstørres ved å zoome inn, men en egen funksjon for å sette tekststørrelse er ikke tilgjengelig. Mister ingen innhold eller funksjonalitet når det er zoomet inn 200%
1.4.5: Images of Text	Result: Passed	Observations: Ingen bruk av bilde av tekst for øyeblikket. Mulighet for at det kommer under "brukerguide", men da er det ikke brukt som eneste måten å få informasjonen, vil også ha forklarende tekst.
1.4.10: Reflow	Result: Passed	Observations: Ingen tap av informasjon eller funksjonalitet.
1.4.11: Non-text Contrast	Result: Passed	Observations: Velg kunnskapsbase-knapper er i kontrast 10.29:1. Ikon i knappene 10.3:1
1.4.12: Text Spacing	Result: Not present	Observations: Ingen innhold bruker markup languages som støtter disse.

1.4.13: Content on Hover or Focus	Result: Passed	Observations: Tooltip tekst på for eksempel hjelp-ikonet nede i høyre hjørne. Teksten dekker ikke til annet innhold, pekeren kan flyttes over teksten uten at den forsvinner, og teksten er synlig frem til sveve- eller fokusutløseren fjernes.
--	----------------	---

2 Operable

2.1 Keyboard Accessible

Success Criterion	Result	Observations
2.1.1: Keyboard	Result: Passed	Observations: Ingen spesifikk timing nødvendig.
2.1.2: No Keyboard Trap	Result: Passed	Observations: Ingen keyboard trap til stede, bruker kan bruke unmodified keys til å komme seg ut av.
2.1.4: Character Key Shortcuts	Result: Not present	Observations: Ingen shortcuts implementert.

2.2 Enough Time

Success Criterion	Result	Observations
2.2.1: Timing Adjustable	Result: Not present	Observations: Ingen tidsbegrensninger er til stede.
2.2.2: Pause, Stop, Hide	Result: Passed	Observations: Tekstvinduet autoscrolls, men brukeren kan enkelt stoppe denne funksjonen.

2.3 Seizures and Physical Reactions

Success Criterion	Result	Observations
2.3.1: Three Flashes or Below Threshold	Result: Not present	Observations: Ingen av sidene har blinkende lys.

2.4 Navigable

Success Criterion	Result	Observations
2.4.1: Bypass Blocks	Result: Not present	Observations: Content er ikke gjentatt på flere sider.
2.4.2: Page Titled	Result: Passed	Observations: Chat-siden krever ingen tittel, men brukerguide siden har tittel som beskriver tema eller hensikt.
2.4.3: Focus Order	Result: Passed	Observations: Ved å "tabbe" seg igjennom siden blir alle elementene fokusert og får ring. Rekkefølger er litt merkelig og går fra valg av kunnskapsbase, til wizzardprompt, så til input før den går til fanen, og URL, og deretter navigasjonsbaren. Men ellers er det mulig å gjør alt av tabbing
2.4.4: Link Purpose (In Context)	Result: Passed	Observations: Teksten for linkene under hjelpesenteret forklarer seg selv, eller sammen med teksten som er programmatically determined.
2.4.5: Multiple Ways	Result: Not present	Observations:

		Såpass få sider at det ikke er nødvendig med site map eller table of contents.
2.4.6: Headings and Labels	Result: Passed	Observations: Headings og labels beskriver tema eller hensikt.
2.4.7: Focus Visible	Result: Passed	Observations: Keyboard focus indicator er synlig.
2.4.11: Focus Not Obscured (Minimum)	Result: Passed	Observations: Når en komponent mottar fokus, er fokusringen fortsatt synlig selv om ikke hele objektet er synlig.

2.5 Input Modalities

Success Criterion	Result	Observations
2.5.1: Pointer Gestures	Result: Passed	Observations: All funksjonalitet som bruker multipoint eller path-based gesture kan utføres med single pointer uten path-based gesture.
2.5.2: Pointer Cancellation	Result: Passed	Observations: Abort or undo muligheter er til stede, eller down-event er essential.
2.5.3: Label in Name	Result: Passed	Observations: Ved å inspisere elementene på siden er det tydelig at de viktigste elementene som brukeren TAR PÅ inneholder labeling, både knapper og bilder. Teksten er meningsfullt og beskriver hva elementet er.

2.5.4: Motion Actuation	Result: Not present	Observations: Ingen handlinger bruker motion detection.
2.5.7: Dragging Movements	Result: Passed	Observations: Handlinger som bruker dragging movements kan også utføres med single pointer uten å dra.
2.5.8: Target Size (Minimum)	Result: Passed	Observations: Størrelsen på målet for pekerinndata er minst 24 x 24 px.

3 Understandable

3.1 Readable

Success Criterion	Result	Observations
3.1.1: Language of Page	Result: Passed	Observations: Ja, default human language for hver side kan bli programmatically determined.
3.1.2: Language of Parts	Result: Passed	Observations: Det menneskelige språket for innholdet i elementet er det samme som det nedarvede språket for elementer som er spesifisert i HTML-attributten lang.

3.2 Predictable

Success Criterion	Result	Observations
3.2.1: On Focus	Result: Passed	Observations: Når en brukergrensesnittkomponent får fokus, initierer den ikke en endring av kontekst.

3.2.2: On Input	Result: Passed	Observations: Endring av innstillingen for en komponent forårsaker ikke automatisk endring av kontekst med mindre bruker er informert. Slik som språk og lysmodus.
3.2.3: Consistent Navigation	Result: Passed	Observations: Brukere forholder seg for det meste til 1 side, men ja navigasjonsmekanismer som gjentas på flere websider forekommer i samme relative rekkefølge hver gang de gjentas.
3.2.4: Consistent Identification	Result: Passed	Observations: Komponenter forholder seg til en felles stil.
3.2.6: Consistent Help	Result: Passed	Observations: Hjelp-mekanismer er til stede. Når de gjentas på andre sider forekommer de i samme relative rekkefølge hver gang de gjentas.

3.3 Input Assistance

Success Criterion	Result	Observations
3.3.1: Error Identification	Result: Passed	Observations: Selve chatbotten vil gi en forklaring dersom den ikke kan svare for eksempel.
3.3.2: Labels or Instructions	Result: Passed	Observations: Input felt har labels eller tooltip.
3.3.3: Error Suggestion	Result: Passed	Observations: Constraints er på plass for å hindre brukefeil. Chat input feltet er for

		eksempel låst frem til brukeren velger hvilken kunnskapsbase de vil kommunisere med.
3.3.4: Error Prevention (Legal, Financial, Data)	Result: Not present	Observations: Bruker har ikke tilgang på slike funksjoner.
3.3.7: Redundant Entry	Result: Passed	Observations: Brukeren har predefinerte valg tilgjengelig.
3.3.8: Accessible Authentication (Minimum)	Result: Passed	Observations: Bruker clerk for innlogging.

4 Robust

4.1 Compatible

Success Criterion	Result	Observations
4.1.2: Name, Role, Value	Result: Passed	Observations: "Note This success criterion is primarily for Web authors who develop or script their own user interface components. For example, standard HTML controls already meet this success criterion when used according to specification." I systemet tar vi i bruk JSX, som er HTML basert. Systemet bruker allerede suksess kriteriene for dette.
4.1.3: Status Messages	Result: Failed	Observations: Failed, skjermleser fungerer på alt, men ved interaksjon feks bytte utseende fra mørk til lys vil ikke skjermleseren fortelle

		at det ble byttet, det samme med slett, send osv
--	--	--

Sample of Audited Web Pages

1. EgdeAI-Learn - <https://learn.ai.egde.no/no>
2. EgdeAI-Learn - <https://learn.ai.egde.no/en>
3. Userguide - https://learn.ai.egde.no/en/user_guide
4. Brukerguide - <https://learn.ai.egde.no/no/brukerguide>

Web Technology

Not provided

Recording of Evaluation Specifics

Not provided

Vedlegg 28 - WAVE evaluering

WAVE web accessibility evaluation tool powered by WebAIM

Styles: OFF ON

Summary

0 Errors	0 Contrast Errors
1 Alerts	4 Features
3 Structural Elements	34 ARIA

View details

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.

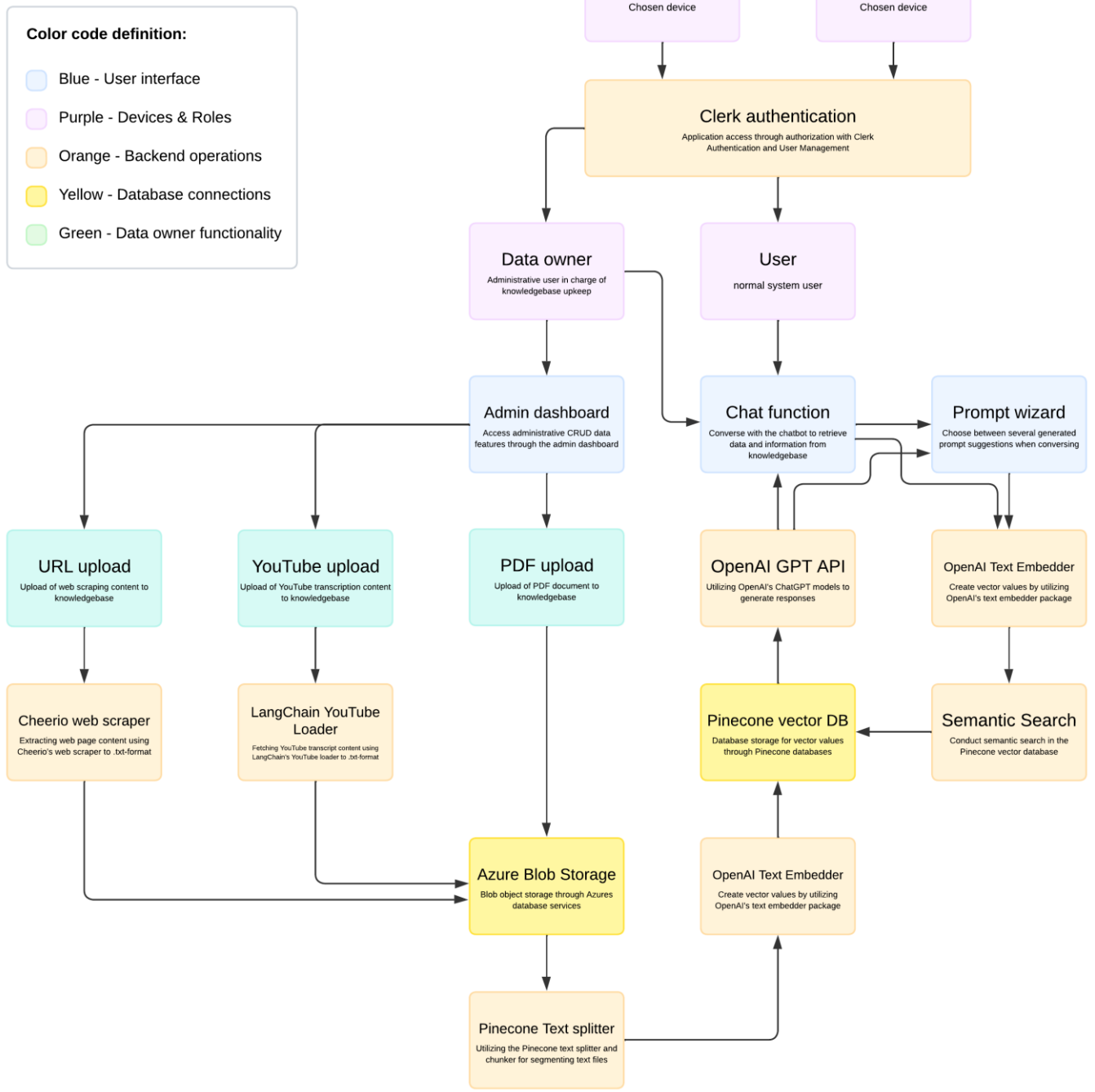
Hei! Velkommen til EgdeAI-Learn, hva har du lyst til å gjøre om i dag?
Vennligst velg tema ved å trykke på en av knappene under for å komme i gang.

- Om Egde
- PersonalHandbok
- Velox
- ÆØÅtest

Velg et av samtaleemne over!

Vedlegg 29 - Systemarkitektur

EgdeAI-Learn System Architecture



Vedlegg 30 - Brukertest

Ønsk velkommen og takk for at de tok seg tid til å delta. Understrek for brukeren at de ikke kan gjøre noe galt, vi vil se om nokke er forvirrende for å kunne forbedre brukeropplevelsen Spør om førsteinntrykk. Be bruker tenkte høyt

desktop og mobil:

- Start en samtale om Alt om Egde
- Fortsett samtalen ved å trykke på tekstfeltet.
- Finn brukerguide
- Finn historikk
- Bytt kunnskapsbase i historikken
- Slett samtalen / endre tittel. (ligger på samme plass. spør noen om slett, andre om å endre tittel for å se om plasseringen føles naturlig ut.
- Start ny samtale
- Logg ut

Spørsmål: om du bruka andre lignende AI-verktøy, hvordan holder du orden i historikk? Sletter du samtaler etter hvert? Bruker du arkiv-funksjonen?

Admin:

- Last opp PDF i kunnskapsbasen Alt om Egde
- Last opp URL i kunnskapsbasen Design
- Sorter dato etter synkende rekkefølge.
- Gjem filtype.
- Vis filtype.
- Vis bare PNG-filer
 - o Ta vekk filteret
- Vis bare filer i kunnskapsbasen Utvikling.
- Slett siste filen.
- Slett alle filer.

Vedlegg 31 - A/B brukertest

Versjon A

The screenshot displays the EgdeAI-Learn chat interface. On the left is a sidebar with a search bar containing "Alt om Egde" and a list of frequently asked questions. The main chat area shows a conversation with an AI assistant. The AI's messages are in white boxes with a purple "AI" icon, and the user's messages are in blue boxes with a user profile icon. The AI provides information about vacation days in Norway, mentioning a minimum of 25 working days per year and advising to check the specific employment contract or contact HR.

EgdeAI Learn

Alt om Egde

Hva driver Egde med?

Hva er Smia?

Hvilke ordninger har Egde for...

Hvor er Egdes kontorer?

Hva bør jeg lære som nyansatt?

Har egde kontaktinformasjon...

Hvordan kan jeg lære mer om...

Hva slags verktøy og systemer...

Hvordan finner jeg en mentor som...

Er det noen spesifikke prosjekter...

Hvordan kan jeg bidra til å styrke...

Hvilke spesifikke spørsmål bør jeg...

Hva bør jeg lære først dersom jeg...

Hvor mange ansatte har Egde?

Hvor ligger hovedkontoret til Egde?

Hva kan du hjelpe meg med?

Hva er Egdes verdier?

Hva kan du hjelpe meg med?

Hvordan logger jeg på printeren?

Hvor mange feriedager har jeg?

Start ny chat

AI: Heil Velkommen til EgdeAI-Learn, hva har du lyst å snakke om i dag?

Vennligst velg tema ved å trykke på en av knappene under for å komme i gang.

Alt om Egde

Utvikling

Design

AI: Så hyggelig at du vil snakke om Egde, hva kan jeg hjelpe deg med?

Hvor mange feriedager har jeg?

AI: Antall feriedager du har avhenger av arbeidskontrakten din og lokale lovkrav. I Norge er det vanlig med minimum 25 virkedager ferie hvert år for fulltidsansatte, i henhold til ferieloven. Det er imidlertid viktig å sjekke din spesifikke arbeidskontrakt eller ta kontakt med HR-avdelingen hos Egde Consulting AS for nøyaktige detaljer angående din personlige ferieavtale.

Skriv her...

Versjon B

EgdeAI² Learn

Alt om Egde

Siste 7 dager

- Hva driver Egde med?
- Hva er Smia?
- Hvilke ordninger har Egde for...
- Hvor er Egdes kontorer?
- Hva bør jeg lære som nyansatt?
- Har Egde kontaktinformasjon...

Siste 14 dager

- Hvordan kan jeg lære mer om...
- Hva slags verktøy og systemer...
- Hvordan finner jeg en mentor som...
- Er det noen spesifikke prosjekter...
- Hvordan kan jeg bidra til å styrke...
- Hvilke spesifikke spørsmål bør jeg...
- Hva bør jeg lære først dersom jeg...
- Hvor mange ansatte har Egde?
- Hvor ligger hovedkontoret til Egde?

Februar

- Hva kan du hjelpe meg med?
- Hva er Egdes verdier?
- Hvordan logger jeg på printerne?
- Hvor mange feriedager har jeg?
- Når ble Egde startet?

Start ny chat

AI: Hei! Velkommen til EgdeAI-Learn, hva har du lyst å snakke om i dag?
Vennligst velg tema ved å trykke på en av knappene under for å komme i gang.

Alt om Egde

Utvikling

Design

AI: Så hyggelig at du vil snakke om Egde, hva kan jeg hjelpe deg med?

Hvor mange feriedager har jeg?

AI: Antall feriedager du har avhenger av arbeidskontrakten din og lokale lovkrav. I Norge er det vanlig med minimum 25 virkedager ferie hvert år for fulltidsansatte, i henhold til ferieloven. Det er imidlertid viktig å sjekke din spesifikke arbeidskontrakt eller ta kontakt med HR-avdelingen hos Egde Consulting AS for nøyaktige detaljer angående din personlige ferieavtale.

Skriv her...

Vedlegg 32 - Referat styringsgruppemøte

Styringsgruppemøte april 2024

Dato: 17.04.24 - I regi av Bachelorgruppen ved Egde Consulting 2024

Oppmøte

Bachelorgruppe:

- Jostein Engesæt (josteinen@uia.no)
- Hassan Heidari (hassanh@uia.no)
- Hans Christian Morka (hanscm@uia.no)
- Frida Lillesalt Valvik (fridalv@uia.no)
- Ivar Michael Tangeraas (ivarmt@uia.no)

Rådgiver fra UiA:

- Hallgeir Nilsen (hallgeir.nilsen@uia.no)

Rådgiver fra Egde:

- Øyvind Brekkhus Sandåker (oyvind.brekkhus.sandaker@egde.no)

Agenda for møte

- Møte starter kl. 12.00 ved Egdes lokaler i Kirkegata 18, 4611 Kristiansand (se *møteinnkalling for grupperom*).
- Kort introduksjon og velkomst til møte.
- Oppsummering av arbeidet så langt:
 - Backlogg
 - Estimeringer
 - Diskusjon og tilbakemeldingsrunde
- Demo (*dersom nødvendig*)
- Plan for videre arbeid:
 - Hva gjenstår
 - Fokusområder
 - Diskusjon og tilbakemeldingsrunde
- Spørsmål og kommentarer
- Møte avsluttes

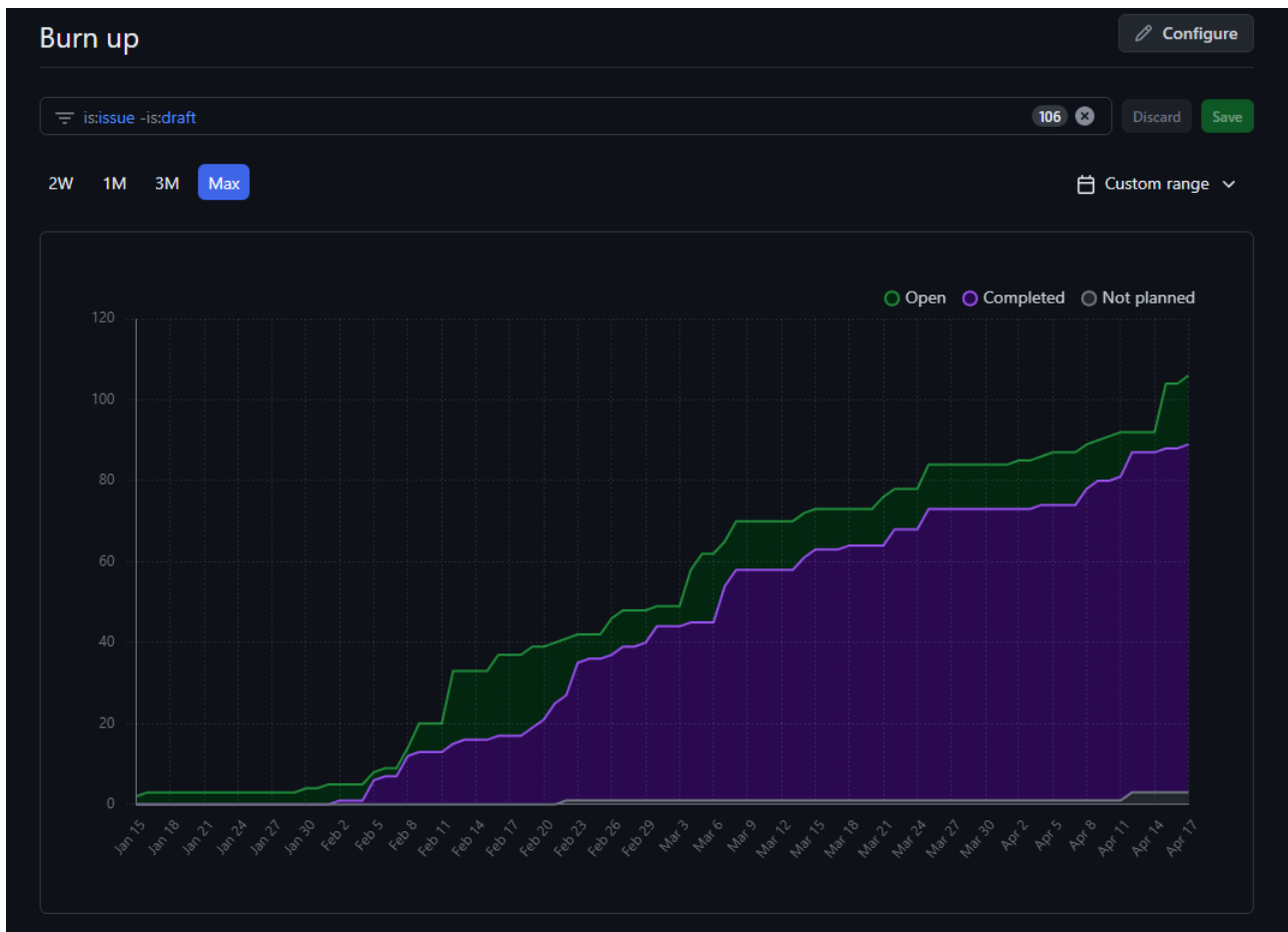
Referat/Oppsummering

Møtet startet som planlagt kl. 12.00 med alle representanter til stede.

Bachelorgruppen starter med å ønske velkommen til møtet og starter med oppsummering av arbeidet så langt.

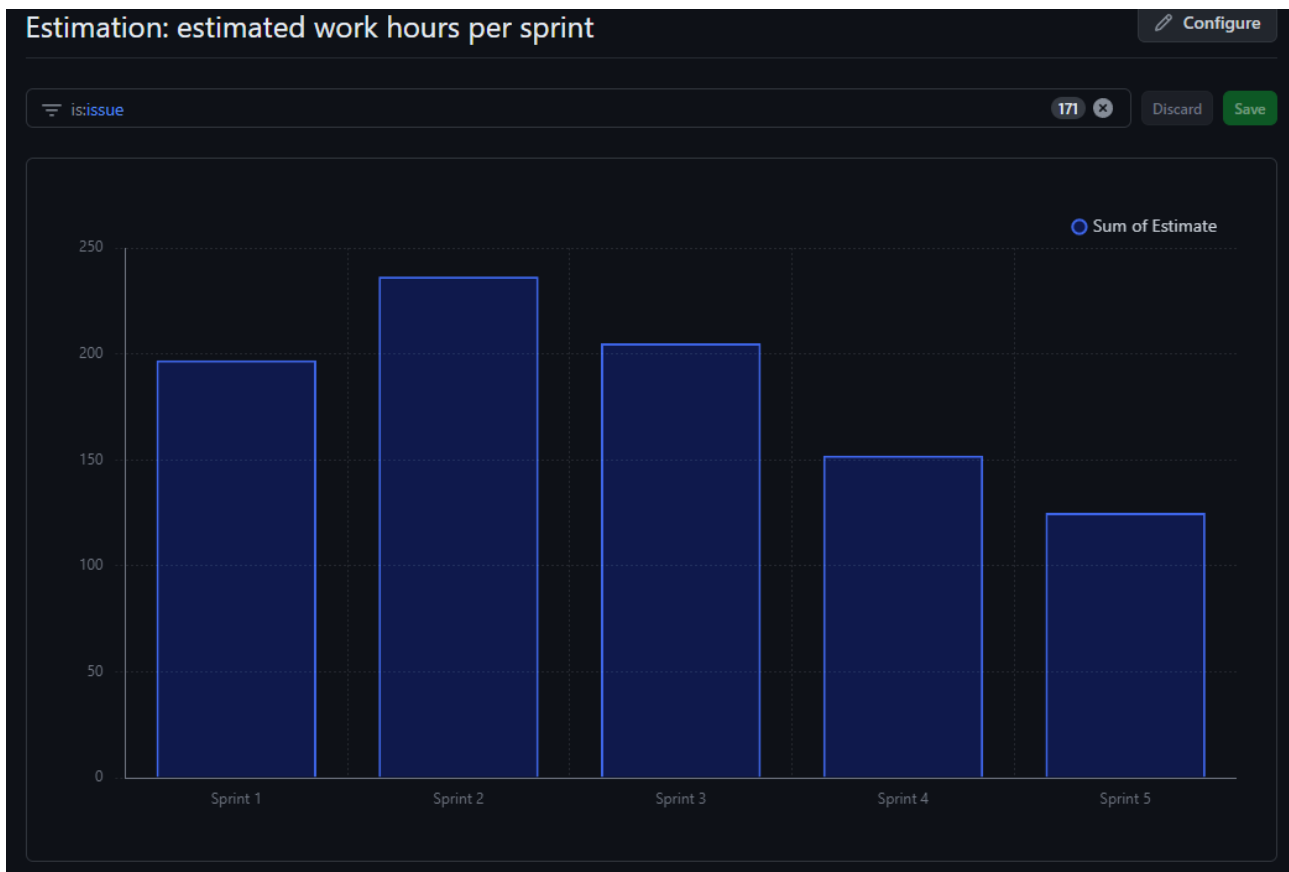
Oppsummeringen baserer seg mye på å forklare arbeidet gjennom utformingen og strukturen av GitHub Projects backloggen som gruppen har brukt gjennom arbeidsløpet frem til nå. For å best

demonstrere og forklare hvordan arbeidsforløpet har vært i prosjektet. Gruppen viser frem blant annet et *burn up* diagram som kan vise til hvordan utformingen av nye backlogg issues og arbeidsoppgaver har blitt utformet over tid, samt hvordan antall fullførte backlogg issues og arbeidsoppgaver følger en stabil linje etter.



Burn up diagram fra januar 15 – april 17

Videre diskuteres arbeid og prosesser rundt estimering av tid rundt arbeidsoppgaver. Gruppen forklare at ved starten av hver sprint utfører gruppen en *planning poker* sesjon, hvor hver arbeidsoppgave diskuteres og får en estimert score for antall estimerte timer arbeidsoppgaven vil kreve fra start til slutt.



Estimasjons diagram fra arbeidsløpet

EgdeAI-Learn

Status Done

Priority High

Size Large

Estimate 13

Work hours 6

Iteration **Sprint 4**
Mar 25 - Apr 12

Start date Apr 5, 2024

End date Apr 5, 2024

Type Backend

Application Learn

The screenshot shows a Jira issue card for "EgdeAI-Learn". The status is "Done" with a green checkmark. The priority is "High" (red triangle), size is "Large" (purple person icon), estimate is 13, and work hours are 6. The issue is assigned to "Sprint 4" which runs from March 25 to April 12, 2024. The start and end dates are both April 5, 2024. The type is "Backend" (yellow pill) and the application is "Learn" (green pill).

Føring i backlogg issue

Gruppen forklarer videre hvordan arbeidstimer blir notert og registrert i arbeidsoppgavene/backlogg issueene underveis i arbeidsløpet. Her blir det diskusjon om gruppen potensielt overestimerer arbeidsoppgaver i stor eller liten grad, og hvordan gruppen dokumenterer arbeidstid som faktisk blir

brukt. Gruppen svarer med at de bevist valgte å heller overestimere arbeidsoppgaver for å ikke falle bak på bestemte milepæler ved hver sprint, og ville heller utnytte gjenværende arbeidstid til andre formål i prosjektarbeidet.

Rådgiver fra UiA etterspør også om estimeringen regnes som *estimate to complete* og om gruppen vurderer riktig bruk av ressurser til gitt estimert arbeidstid. Gruppen legger frem at estimeringen beregnes fra oppstart av arbeidsoppgave til slutt, som defineres som at en gitt oppgave vurderes som komplett/ferdig og klar til leveranse. Gruppen legger til at arbeidstid og ressurser som blir til overs basert på utført arbeidstid blir fordelt på andre arbeidsoppgaver og rapportskrivning.

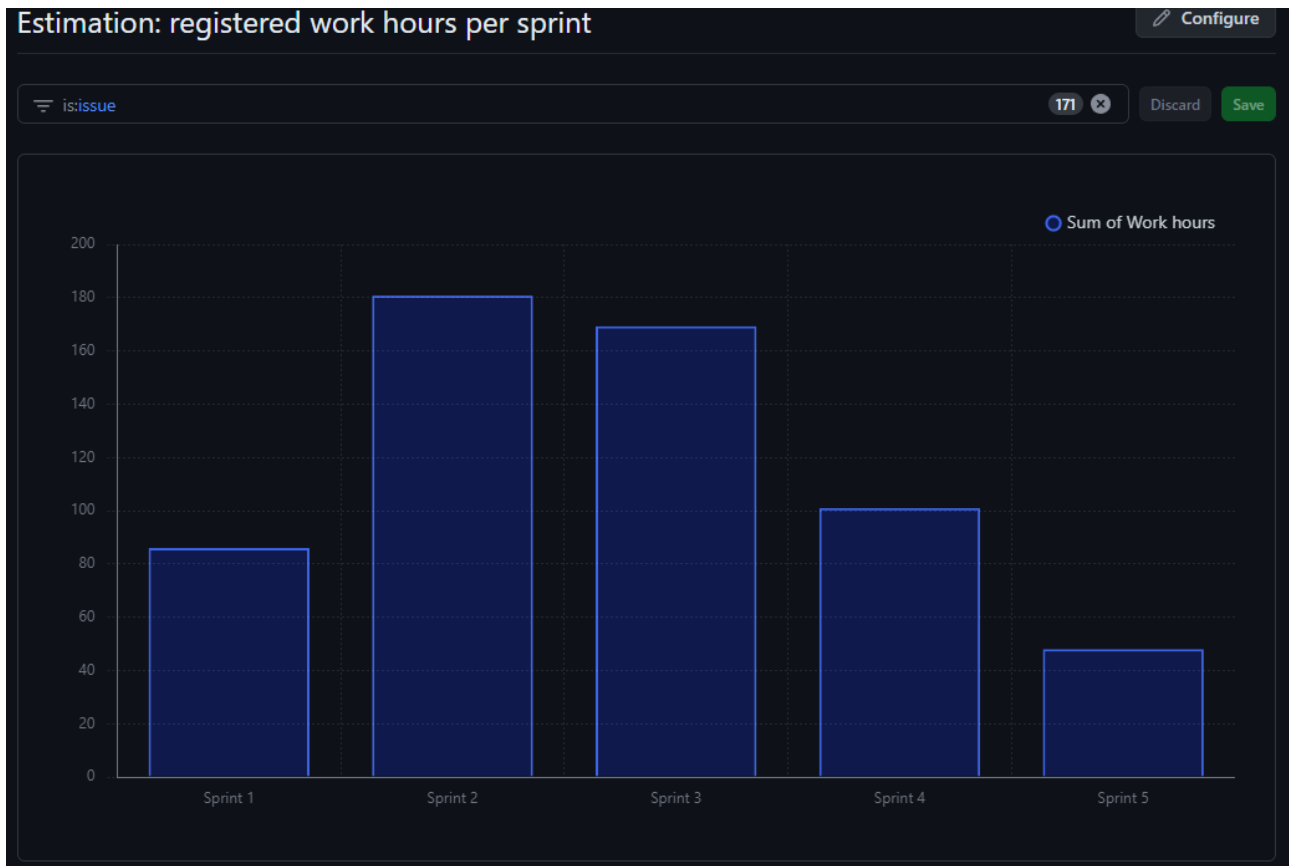


Diagram for registrert arbeids fra arbeidsløpet

Videre viser gruppen til et diagram som viser oversikt over mengde gitte oppgaver fra hvert fagområde (backend, dokumentasjon, osv.) Her forklares og illustreres kurven av endringer gjennom arbeidsprosessen hvor noen fagområder har spilt større roller i noen sprints, for så å ha større påvirkning senere i arbeidsløpet.

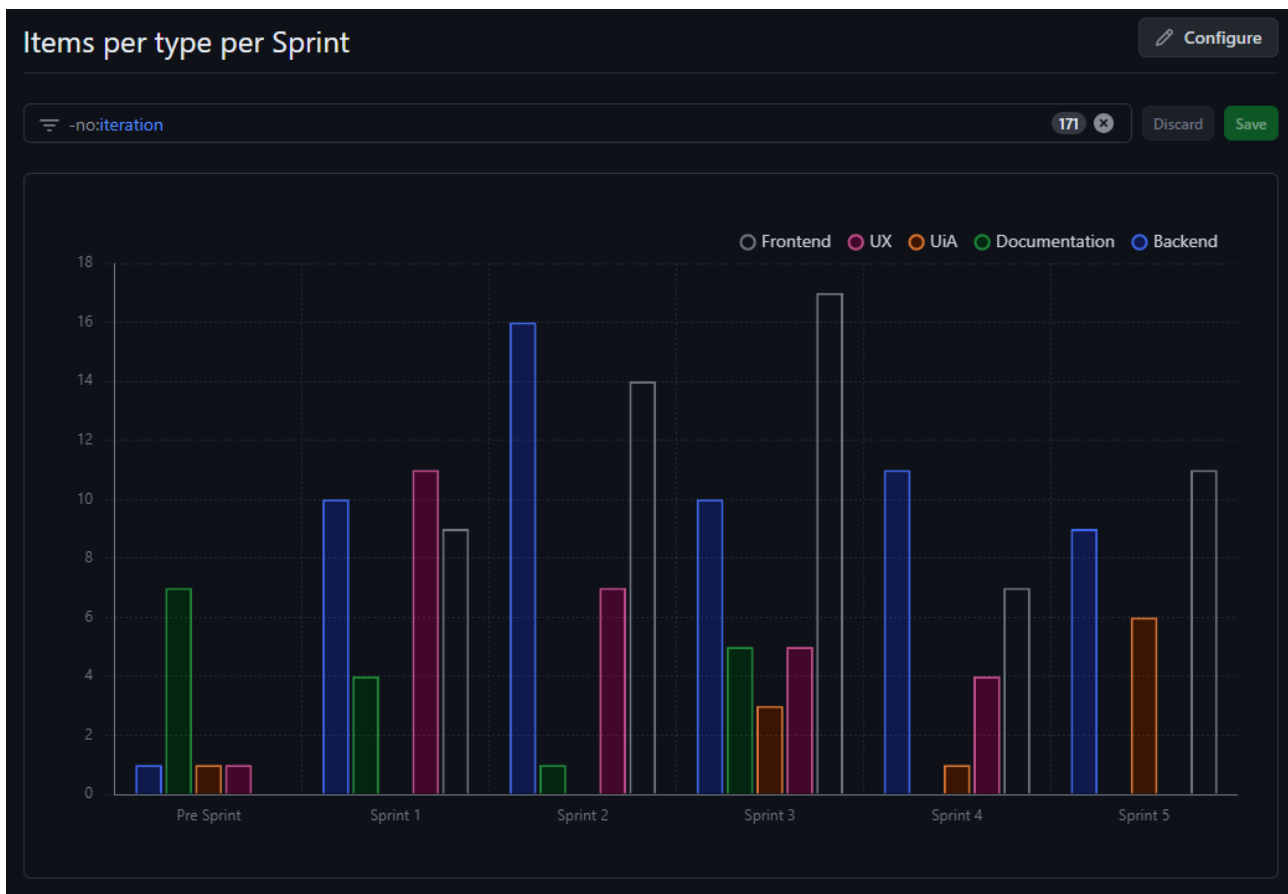


Diagram for oversikt over fagområder gjennom sprintene

Gruppen går videre med å vise til diagrammer som illustrerer bevegelsesmønsteret til backlogg issues gjennom sprintene. Her ble det vist frem ett diagram per sprint.



Diagram for backlog issue flyt gjennom sprint #2

Oppsummering av utført arbeidsprosess avsluttes med en gjennomgang og forklaring av backlogg issue føring. Gruppen viser til hvordan valgte arbeidsprosesser blir nøye dokumentert og oppført per issue. Dette begrunnes med at dokumentasjonen kan bidra til innsikt i arbeidsprosessen og valg ved senere anledning, egen refleksjon på arbeid, og ikke minst synlighet og forståelse for de andre medlemmene i scrum-teamet.

FE | Add code blocks to chat-response #295

Open EgdeConsulting/egde-ai Private

imttv opened 2 days ago edited by imttv · Edits · ...

The chatbot can respond with code and other code-block relevant output, but the format for displaying the values can be better with proper code-block syntax.

Hvordan kan jeg bruke useParams sammen med useEffect for å hente data basert på URL-parametere?

For å hente data basert på en URL-parameter i en React-komponent, kan du kombinere useParams hook fra react-router-dom med useEffect og useState hooks fra React. Dette mønsteret er nyttig når du trenger å utføre en sideeffekt, som å hente data fra en API, basert på verdien av en URL-parameter. Her er et eksempel på hvordan dette kan gjøres:

- 1. Definer en rute med en parameter:** Sørg for at du har en rute definert i din applikasjon som inkluderer en parameter. For eksempel, en rute for å vise detaljer om en bruker basert på deres ID.

```
<Route path="/bruker/:userId"> <UserDetails /> </Route>
```

- 2. Bruk useParams for å hente parameteren:** I UserDetails-komponenten, bruk useParams for å hente den aktuelle userId fra URL-en.
- 3. Kombiner useParams med useEffect for å hente data:** Bruk useEffect for å reagere på endringer i userId og hente data basert på denne verdien. Du kan også bruke useState for å lagre dataen som hentes.

Her er et eksempel på hvordan UserDetails-komponenten kan implementeres:

```
import React, { useState, useEffect } from 'react'; import { useParams } from 'react-router-dom';
function UserDetails() { const { userId } = useParams(); // Hent userId fra URL const [userDetails, setUserDetails] = useState(null); // State for å lagre brukerdetaljer useEffect(() => { // Funksjon for å hente brukerdetaljer basert på userId const fetchUserDetails = async () => { try { const
```

Assignees: imttv

Labels: enhancement, learn

Projects: EgdeAI-Learn

Status: In progress

Priority: Medium

Size: Medium

Estimate: 5

Work hours: 8

Iteration: Sprint 5 • Apr 15 - May 3 • Current

Start date: No date

End date: No date

Type: Frontend

Application: Learn

Milestone: No milestone

Skjermdump av et backlogg issue fra sprint #5

Helt til slutt, avsluttes oppsummeringen av arbeidsprosessen med diskusjon om hva som har blitt fremvist og diskutert. Her blir blant annet tematikken rundt utformingen av produkt backlogg diskutert. Gruppen forklarer at produkt backloggen spiller en annen rolle i deres prosjektgjennomføring. Da bestilling fra kunde i dette tilfelle var rimelig åpent og ingen klar produktspesifikasjon var gitt, mente gruppen selv at det var mer naturlig å danne en grunnidé for hva sluttproduktet/forskningen kunne være. Med dette i bakhodet, valgte gruppen bestemt at en mer agil tilnærming til utforming av produkt backloggen ville skje via utformingen av sprint backloggene i løpet av prosjektet. Det ga mulighet for rask endring og utforming av produktspesifikasjon hele veien gjennom prosjektløpet. Denne diskusjonen bidro til å forstå arbeidsstrukturen til bachelorgruppen opp mot backloggen.

Her poengteres det også fra rådgiverne at det blir viktig for gruppen å kunne definere hvorfor valgene ble tatt, men ikke minst resultatet av valgene. *“Hvorfor var det lurt å gjøre det på denne måten?”* Det blir også gitt en oppfordring om å spille litt på denne manglene klare produktbestilling i bachelorrapporten for å fremheve akkurat hvorfor denne tilnærmingen fungerte for dette prosjekt. Videre i møtet blir det bestemt at tidsrommet for resten av møtet ikke gir rom for visning av demo, men dette begrunnes med at begge rådgiverne er mer eller mindre oppdatert på de siste funksjonalitetene og løsningen til systemet.

Som avslutning på møte legger gruppen frem sine planer for videre arbeid opp mot deadline for bachelorrapport levering 16. Mai. Det legges frem at gruppen planlegger en strategisk *code freeze* i slutten av nåværende sprint (sprint #5) på følgende datoer.

- Soft freeze: 26. April
- Hard freeze: 6. Mai

Her poengteres det fra rådgiver fra UiA om at videre arbeid med kode kan fint skje etter bachelorrapportens deadline. Gruppen svarer med at slik prosjektløpet er satt opp i dag er målet å være ferdig med all form for kode-implementasjon før deadline for bachelorrappport innlevering.

Videre foregår samtalen løst om arbeidet, Egdes syn på innsats og resultat, forventinger og tips til eksamen.

Møtet avsluttes kl. 13:00 som planlagt.