

Forprosjektrapport  
Url-forkorter for Nav AS



VÅR 2025  
Bacheloroppgave DATA3900

**Gruppe 20**

Lara Hassanieh (s374210)  
Mahnoor Kamran Sadiq (s374169)  
Kathrine Hassanieh (s374934)  
Sigurd Omnes (s356213)

## Innholdsfortegnelse

### **1. Presentasjon**

- 1.1 Gruppemedlemmer
- 1.2 Oppdragsgiver og kontaktpersoner
- 1.3 Veileder

### **2. Sammendrag**

### **3. Dagens situasjon**

### **4. Mål og rammebetingelser**

### **5. Løsninger og alternativer**

- 5.1 Foreslått løsning
- 5.2 Alternative løsninger
- 5.3 Analyse av virkninger

### **6. Arbeidsplan**

- 6.1 Fremdriftsplan
- 6.2 Milepæler

### **7. Konklusjon**

# 1. PRESENTASJON

## Gruppemedlemmer

Lara Hassanieh, dataingeniør  
[S374210@oslomet.no](mailto:S374210@oslomet.no)

Mahnoor Kamran Sadiq, dataingeniør  
[S374169@oslomet.no](mailto:S374169@oslomet.no)

Kathrine Hassanieh, informasjonsteknologi  
[S374934@oslomet.no](mailto:S374934@oslomet.no)

Sigurd Ivar Volden Omnes, dataingeniør  
[S356213@oslomet.no](mailto:S356213@oslomet.no)  
**Oppdragsgiver**

NAV AS  
Arbeids- og velferdsdirektoratet  
Fyrstikkalleen 1, 0661 Oslo

## Kontaktperson oppdragsgiver

Kyrre Havik  
Utvikler

[Kyrre.Havik@nav.no](mailto:Kyrre.Havik@nav.no)

+47859817

## Internveileder ved Oslomet

Andrea Arcuri  
Professor

Fakultet for teknologi, kunst og design  
Institutt for informasjonsteknologi  
Skyteknologi, datasikkerhet og programutvikling

[Andrea.arcuri@oslomet.no](mailto:Andrea.arcuri@oslomet.no) / [arcuri82@gmail.com](mailto:arcuri82@gmail.com)

+47 67237190

**Om oppdragsgiver: NAV AS**

NAV AS er en nøkkelaktør i den norske velferdsforvaltningen og spiller en sentral rolle i å sikre sosial trygghet og økonomisk bærekraft for innbyggerne. NAV administrerer kritiske tjenester som sykepenger, arbeidsledighetstrygd, pensjon og andre velferdsytelser, samtidig som de aktivt bidrar til arbeidsinkludering og oppfølging av arbeidssøkere.

Med visjonen "muligheter for alle" samarbeider NAV med både offentlige og private aktører for å utvikle innovative løsninger som styrker evnen til å tilby effektive og brukervennlige tjenester. Gjennom kontinuerlig utvikling av teknologiske løsninger ønsker NAV å forbedre tilgangen til informasjon og optimalisere ressursutnyttelsen i sine systemer.

I dette prosjektet samarbeider vi med NAV for å utvikle en intern tjeneste for forkorting av nettadresser. Denne løsningen skal møte spesifikke behov som sikker deling av informasjon og brukervennlig navigasjon i komplekse systemer. Prosjektet inkluderer å designe og implementere en skalerbar løsning som kan utvides til ekstern bruk, med fokus på sikkerhet og effektivitet.

## 2. Sammendrag

Dette prosjektet tar sikte på å utvikle en intern tjeneste for NAV som forkorter lange nettadresser, tilpasset organisasjonens spesifikke behov. Hovedmålet er å forenkle deling av informasjon internt, med en løsning som kan utvides for fremtidig ekstern bruk. Tjenesten skal oppfylle krav til GDPR-samsvar, sikkerhet, og brukervennlighet.

Løsningen vil inkludere funksjoner som generering og administrasjon av korte nettadresser, visning av bruksstatistikk, validering av lenkers destinasjon, samt sletting og redigering av eksisterende lenker. Gjennom Test Nais-plattformen vil prosjektet dra nytte av eksisterende teknisk infrastruktur, noe som bidrar til effektiv utvikling og integrasjon.

Prosjektet fokuserer på sikkerhet, skalerbarhet, og fleksibilitet, og vil implementeres med React for front-end og Kotlin for back-end. Dette sikrer en robust og moderne applikasjon som møter NAVs krav. Fordelene inkluderer økt effektivitet, bedre kontroll over delte data, og en mer strømlinjeformet arbeidsflyt. Utfordringer som læringskurver for nye teknologier og avhengighet av plattformen adresseres gjennom nøye planlegging og kontinuerlig evaluering.

Ved å møte NAVs behov for bedre deling av informasjon, vil prosjektet bidra til økt produktivitet og sikkerhet i organisasjonen. Løsningen legger grunnlaget for fremtidig innovasjon og kan videreutvikles for å styrke NAVs digitale tjenester, både internt og eksternt.

### 3. Dagens Situasjon

NAV bruker flere interne systemer for informasjonsdeling, men mange genererer lange og komplekse nettadresser som er vanskelig å dele og bruke effektivt. SharePoint er et typisk eksempel, der uleselige HTML-entities gjør lenkene upraktiske, særlig i muntlig kommunikasjon. Dette skaper frustrasjon blant ansatte og hemmer en effektiv arbeidsflyt. Offentlige forkortelsestjenester som Bit.ly og TinyURL oppfyller heller ikke NAVs krav, da de mangler sikkerhet og kontrollmuligheter.

Behovet for en intern løsning som sikrer brukervennlighet, fleksibilitet og full kontroll er derfor kritisk. En slik løsning vil styrke NAVs evne til å administrere og dele lenker på en sikker måte, samtidig som den forenkler arbeidsprosessene. Den kan også utvides til ekstern bruk i fremtiden, noe som vil styrke NAVs digitale kommunikasjonsstrategi og legge til rette for en mer effektiv informasjonsflyt både internt og eksternt. Dette prosjektet skal møte disse behovene gjennom en skreddersydd plattform som oppfyller NAVs krav til sikkerhet og brukervennlighet.

## 4. Mål og Rammebetingelser

Målet med dette prosjektet er å utvikle en intern tjeneste for NAV som forkorter lange nettadresser på en måte som oppfyller organisasjonens spesifikke behov. Tjenesten skal forenkle deling av informasjon, forbedre brukervennlighet og effektivitet, samt gi full kontroll over lenkene som deles. Løsningen skal også være skalerbar, med mulighet for fremtidig ekstern implementering, og oppfylle krav til GDPR-samsvar og datasikkerhet.

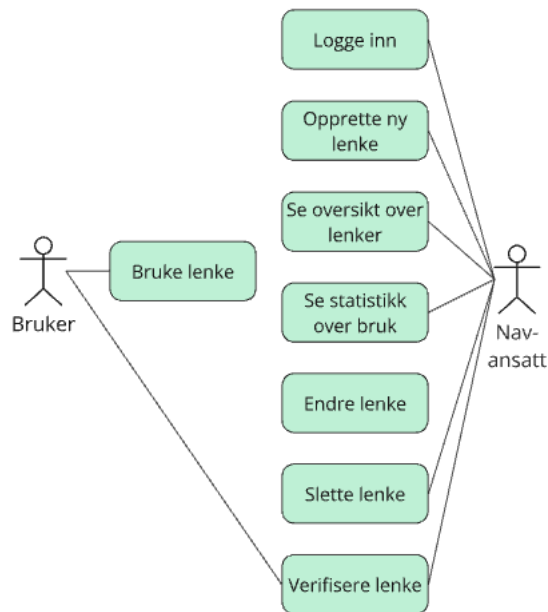
For å oppnå dette er det essensielt at tjenesten gir brukerne mulighet til å generere korte lenker både automatisk og ved tilpasning, administrere eksisterende lenker, vise statistikk for bruk og validere lenkers destinasjon uten innlogging. Effektiviteten i tjenesten er en nøkkelprioritet, og systemet skal kunne generere korte lenker på under 500 millisekunder i minst 95 % av tilfellene. Videre skal det kunne håndtere minst 1000 samtidige brukere uten merkbare ytelsesproblemer. Brukervennlighet står også sentralt, og tjenesten skal designes med et intuitivt grensesnitt, testet gjennom brukertesting med ansatte i NAV for å sikre at minst 85 % vurderer tjenesten som enkel å bruke.

Rammebetingelsene for prosjektet inkluderer teknologiske og organisatoriske krav. Løsningen skal utvikles med React for front-end og Kotlin for back-end, med integrasjon i Test Nais-plattformen. Utviklingsperioden strekker seg til 31. mai 2025, med milepæler som dekker design, implementering, testing og ferdigstilling. Begrensninger inkluderer manglende direkte tilgang til NAVs interne systemer, noe som betyr at all utvikling og testing må foregå i Test Nais-sandkassen.

Sikkerhet og personvern er avgjørende for prosjektet. Tjenesten skal tilfredsstille alle krav til GDPR, inkludert sikker lagring og håndtering av data. All administrasjon av lenker vil være autentisert for å sikre at bare autoriserte brukere kan gjøre endringer. I tillegg vil prosjektet implementere logging av alle relevante operasjoner for å opprettholde datasikkerhet og sporbarhet. Til slutt skal systemet gi tydelig statistikk for bruk av korte lenker, slik at brukerne kan måle effektiviteten og sikre kontroll over delte data.

Denne løsningen forventes å forbedre NAVs interne arbeidsflyt betydelig, samtidig som den legger grunnlaget for fremtidige utvidelser. Med fokus på brukervennlighet, sikkerhet og

effektivitet, vil prosjektet bidra til å styrke NAVs digitale kommunikasjonsstrategi og optimalisere informasjonsdeling i organisasjonen.



## 5. Løsninger og alternativer

Valg av teknologi for dette prosjektet er nøye vurdert for å sikre best mulig samsvar med NAVs eksisterende infrastruktur og krav til effektivitet, skalerbarhet og brukervennlighet. Etter en analyse av tilgjengelige alternativer har vi valgt å bruke Kotlin for back-end og React for front-end, da disse løsningene best imøtekommer behovene for prosjektet.

Kotlin er det mest brukte programmeringsspråket i NAVs interne systemer, og det har blitt foretrukket for utvikling på grunn av sin kompatibilitet med Java Virtual Machine (JVM) og eksisterende Java-kodebaser. Kotlin gir en mer konsis syntaks sammenlignet med Java, noe som reduserer utviklingstiden og gjør koden lettere å vedlikeholde. Videre tilbyr Kotlin støtte for både objektorientert og funksjonell programmering, noe som gir fleksibilitet i utformingen av løsningen. Bruken av Kotlin gir også fordelen av at NAVs utviklingsteam allerede har omfattende erfaring med språket, noe som muliggjør sømløs integrasjon med eksisterende systemer og reduserer behovet for opplæring.



Alternativer som Java ble vurdert, men forkastet fordi Kotlin gir en mer moderne tilnærming med bedre utvikleropplevelse og raskere implementering. Andre backend-alternativer som .NET ble også vurdert, men valgt bort da NAV i liten grad bruker denne teknologistakken internt. Bruken av Kotlin på Test Nais-plattformen sikrer kompatibilitet og stabilitet, og gir tilgang til verktøy og tjenester som allerede er tilpasset NAVs behov.

På front-end-siden ble React valgt som rammeverk for utviklingen av brukergrensesnittet. Reacts komponentbaserte struktur gjør det enkelt å bygge skalerbare og dynamiske grensesnitt, noe som er viktig for dette prosjektet. Videre er React godt støttet i utviklermiljøet, med et stort økosystem av tredjepartsbiblioteker som kan akselerere utviklingsprosessen. Selv om noen teammedlemmer har begrenset erfaring med React, er rammeverket et naturlig valg fordi NAV allerede bruker JavaScript som primær front-end-teknologi, og React er blant de mest populære biblioteker i dette økosystemet.

Alternativer som Angular og Vue.js ble vurdert, men valgt bort. Angular er et mer omfattende rammeverk som kan være overveldende for prosjektets skala, og Vue.js har mindre støtte og fellesskap sammenlignet med React. Reacts fleksibilitet og popularitet gjør det derfor til det beste valget for dette prosjektet.

Til sammen gir valget av Kotlin og React en optimal balanse mellom effektivitet, skalerbarhet og kompatibilitet med NAVs eksisterende systemer. Dette reduserer risikoen for tekniske utfordringer og sikrer at prosjektet kan leveres innen tidsfristen, samtidig som det legger grunnlaget for fremtidig vedlikehold og utvidelse.

#### *- Analyse av virkninger*

Den tekniske løsningen som er skissert for URL-forkortingstjenesten i Test Nais har flere viktige implikasjoner for både brukere og den generelle systeminfrastrukturen.

Implementeringen av tjenesten vil direkte adressere Navs behov for et funksjonelt og sikkert verktøy som lar brukere forkorte URL-er, administrere opprettede adresser og se statistikk om bruken deres. Nedenfor analyserer vi de potensielle effektene av å implementere denne løsningen, både fra et brukerperspektiv og et teknisk perspektiv.

#### *Brukerpåvirkning*

### **Forbedret effektivitet og brukeropplevelse**

Tjenesten gir brukerne mulighet til å forkorte og tilpasse URL-er, noe som forenkler deling og øker brukervennlighet. Ansatte i NAV vil enkelt kunne generere kortere, mer håndterbare URL-er, noe som gjør arbeidsflyten smidigere, spesielt for lange eller komplekse adresser. Tilpassede adresser gir et ekstra lag av personalisering, som gjør dem enklere å huske og dele.

### **Tilgangskontroll og sikkerhet**

Ved å kreve autentisering for opprettelse og administrasjon av URL-er, sikrer systemet høy sikkerhet og kontroll. Et personlig dashboard lar brukerne administrere sine URL-er, inkludert sletting og redigering, med en revisjonssporing for å opprettholde integriteten til delte koblinger.

### **Brukergjennomsiktighet og URL-statistikk**

Statistikkfunksjonen gir brukerne verdifull innsikt i hvor ofte URL-er brukes, noe som kan være nyttig for å måle engasjement i delte ressurser som rapporter og presentasjoner. Videre kan brukere validere destinasjonen til en URL uten å logge inn, noe som øker tilliten og reduserer risikoen for ondsinnede eller villedende lenker.

### **Administrasjon og fleksibilitet**

Tjenesten gir brukerne mulighet til å administrere innholdet sitt effektivt, inkludert sletting av foreldede eller feilaktige lenker. Selv om adresseendring ikke er prioritert, kan dette vurderes i fremtidige iterasjoner for økt fleksibilitet.

### *Teknisk innvirkning*

#### **Autentisering og brukertilgang**

Implementeringen av et påloggingssystem legger til et nivå av kompleksitet, da det krever sikker håndtering av autentisering og brukerøker. Fordelene med dette inkluderer kontroll over brukeroperasjoner og bedre administrasjon av URL-er, noe som oppveier de tekniske utfordringene.

### **Generering og tilpasning av adresser**

Backend må støtte både automatisk og tilpasset generering av URL-er, med sanntidssjekker for tilgjengelighet. Dette øker systemets kompleksitet, men sikrer unike og brukervennlige lenker.

### **Databaseadministrasjon og dataintegritet**

For å lagre informasjon om opprettede URL-er og tilhørende statistikk, kreves en godt strukturert database. Systemet må kunne håndtere store datamengder effektivt, med mekanismer for spørring, oppdatering og logging for å sikre dataintegritet.

### **Statistikksporing og rapportering**

Sanntidssporing av URL-bruk innebærer registrering av klikkdata og andre metadata. Effektiv caching og dataaggregering er nødvendig for å sikre rask statistikktilgang uten ytelsestap, spesielt ved stor skala.

### **Internt vs. eksternt adressebytte**

Selv om støtte for interne og eksterne adresser ikke er prioritert, må systemet kunne skille mellom de to typene ved fremtidig utvidelse. Dette kan innebære å utvikle spesifikke ruting- og valideringslogikker.

## 6. Framdriftsplan

