

# Visjonsdokument

## System for kommunal kommunikasjon

### Anvendt Informasjonsteknologi, IDI, NTNU



# HverdagsHelt

#### Revisjonshistorikk

Dato	Versjon	Beskrivelse	Forfatter
20/11/18	1.0	Førsteutkast	Emil Eidesmo, Mathias Bytingsvik, Tommy Haukås, Frida Magnarsdatter Lund, Knut Yngve Barstad Gjelle
03/01/19	1.1	Endret ikke-funksjonelle krav	Nils Tesdal
06/01/19	1.2	Oppdatert etter tilbakemeldinger	Emil Eidesmo, Mathias Bytingsvik, Tommy Haukås, Frida Magnarsdatter Lund, Knut Yngve Barstad Gjelle

# Innholdsfortegnelse

<b>1 Innledning</b>	<b>3</b>
<b>2 Visjon for systemet</b>	<b>3</b>
<b>3 Sammendrag problem og produkt</b>	<b>3</b>
3.1 Problemsammendrag	3
3.2 Produktsammendrag	4
<b>4 Overordnet beskrivelse av interessenter og brukere</b>	<b>4</b>
4.1 Oppsummering interessenter	4
4.2 Oppsummering brukere	5
4.3 Brukermiljøet	5
4.4 Sammendrag av brukernes behov	6
4.5 Alternativer til vårt produkt	7
<b>5 Produktoversikt</b>	<b>8</b>
5.1 Produktets rolle i brukermiljøet	8
5.2 Forutsetninger og avhengigheter	8
<b>6 Produktets funksjonelle egenskaper</b>	<b>8</b>
6.1 Alternativer til utvidelser	8
<b>7 Ikke-funksjonelle egenskaper og andre krav</b>	<b>9</b>
7.1 Forslag til WIKI-struktur	10
7.1.1 Domenemodell	10
7.1.2 User stories	10
7.1.3 Brukertester/konseptuelle modeller	11
7.1.4 Arkitektur	11
7.1.5 Prosjektstruktur	11
7.1.6 Klassediagram	11
7.1.7 Server-tjenester	11
7.1.8 Databasemodell	11
7.1.9 Dokumentasjon av kildekode	11
7.1.10 Installasjon og kjøring	11
7.1.11 Kontinuerlig integrasjon og testing	11
7.1.12 Sikkerhet	11

## 1 Innledning

Dette dokumentet beskriver overordnede krav til prosjektoppgaven i emnet Systemutvikling 2 med webapplikasjoner. Oppgaven består i å utvikle webapplikasjonen HverdagsHelt, et system med oversikt over gjøremål i kommunen og et system som gir brukerne muligheten til å varsle om nye mangler. Prosjektet skal utføres av 2. år dataingeniør i løpet av starten av vårsemester i 2019.

## 2 Visjon for systemet

“Enklere kommunikasjon med DIN kommune”.

HverdagsHelt skal gjøre det enkelt for innbyggerne i kommune Norge og varsle om feil og mangler på infrastruktur, samt gi brukerne informasjon om ulike hendelser og events i kommunen. Systemet skal være web-basert og bruk fra mobile enheter sees på som sentralt.

## 3 Sammendrag problem og produkt

### 3.1 Problemsammendrag

HverdagsHelt skal være en plattform for alle kommuner i Norge hvor man kan melde inn om feil og mangler ved infrastruktur i de ulike tettstedene. I dag finnes det utallige løsninger i de ulike kommunene og ingen felles plattform. Det er også tettsteder som mangler et slikt knutepunkt og som kun har et telefonnummer som en kontaktmulighet. HverdagsHelt skal gjøre det enklere og raskere å melde fra om feil og mangler på blant annet veisystemer, friarealer, vann og avløp uten å kreve mer fra kommunene enn dagens løsninger.

Problem med	Rapportering av problemer på infrastruktur i kommunenorge er forskjellig fra kommune til kommune, og ofte vanskelig å finne.
Berører	Alle som tar seg tid til å rapportere om problemer eller leter etter informasjon.
som resultatet av dette	Dagens systemer er ineffektive og demotiverende for innbyggerne å bruke.
en vellykket løsning vil	Gjøre det enkelt å: <ul style="list-style-type: none"><li>• rapportere feil ved infrastrukturen.</li><li>• finne informasjon om events og hendelser.</li></ul>

### 3.2 Produktsammendrag

Plattformen kan brukes av privatpersoner, ansatte i kommunen eller fylket, og kan også brukes av private bedrifter (som feks SmartPark). Dette skjer via et innloggingssystem. HverdagsHelt fungerer også som en informasjonskanal og varslingssystem til de som har opprettet en bruker, man kan selv velge hva slags informasjon man blir varslet om. Systemet skal kunne brukes uavhengig av hvor man er i Norge, men den vil ta i utgangspunkt i din hjemkommune, og det er derfra du vil få informasjon og varsler.

For	beboere, virksomheter og ansatte i kommunen.
HverdagsHelt	En webapplikasjon som er tilgjengelig på tvers av plattformer og nettlesere.
Som	Varsling fra borger til kommune om feil, skader, graffiti ol. Men også varsling til brukere om sin melding, eller annen informasjon den er interessert i.
I motsetning til	At beboere må ringe inn, melde fra manuelt eller at det foreligger store variasjoner mellom hver kommune.
Er det nye produktet	Webbasert, multiplattform, mer effektiv, likt mellom kommuner, tilrettelegger for kommunikasjon og samfunnsengasjerte beboere.

## 4 Overordnet beskrivelse av interessenter og brukere

### 4.1 Oppsummering interessenter

Her beskriver vi de ulike interessenten og hva slags rolle de har i forhold til HverdagsHelt, og også hvilken rolle de skal ha under utviklingen av systemet

Navn	Utdypende beskrivelse	Rolle
Bruker	Sluttbruker av systemet	Registrere seg selv som bruker. Melde fra om feil ved infrastruktur. Få oversikt over andre meldte feil. Ta bilde og plassere hendelse på kart.
Veiledere	Faglærere	Scrummaster, veilede ved utvikling av system.
Prosjektgruppe	Utviklere av systemet	Administrasjon og vedlikehold
Oppdragsgivere (HverdagsHelt)	Prosjekteiergruppe	Produkteier

Anvendt informasjonsteknologi, IDI, NTNU	Økonomisk støtte til maskinvare og eventuelt nødvendig programvare	Administrasjon og vedlikehold
--	--	-------------------------------

## 4.2 Oppsummering brukere

Navn	Utdypende beskrivelse	Rolle under utviklingen	Representert av
Bruker	Bruker som registrerer seg i systemet.	Testing	Utviklerne, eventuelt eksterne personer
Administrator	Opprette kategorier for feil.	Testing	Utviklerne
Privatperson	Kan være hvem som helst	Testing	Utviklerne
Kommuneansatt	Ansatt i kommune som har ansvar for å respondere på henvendelser	Testing	Utviklerne
Bedriftsbruker	Bruker med begrenset mulighet til å respondere på enkelte henvendelser (F.eks kan en entreprenør oppdatere et varslet hull når det er fikset	Testing	Utviklerne

## 4.3 Brukermiljøet

Alle brukere bruker løsningen med valgfri nettleser både på PC og mobil, derfor må plattformen fungere i de vanligste nettleserne og være optimalisert for mobilvisning og navigering.

#### 4.4 Sammendrag av brukernes behov

Behov	Prioritet	Vedrører	Dagens Løsning	Foreslått løsning
Registrere bruker	Høy	Innlogging	Ingen	Selvregistrering der bruker velger brukernavn og passord
Endre passord	Lav	Innlogging	Ingen	Ved innlogging kan bruker endre passord
Melde om feil og mangler	Høy	Rapportering	Varierer fra kommune til kommune	Bruker velger sted via googlemaps eller annet kartverktøy/API og forklare hva som er galt med tekst og eventuelt bilde av stedet.
Bildefunksjonalitet til rapportering.	Middels	Rapportering	Ingen	Mulighet for bruker til å også legge til et bilde i feilmeldingen. Fra mobil.
Sette status til en mangel	Middels	Rapportering	Varierer fra kommune til kommune	Administrator "svarer" på en melding om feil å setter en status på oppgaven.
Tilbakemelding til melder om status på problemet	Middels	Rapportering	Ingen	Bruker skal få en oppdatert status på problemet: Mottatt, under saksbehandling, fullført.
Slette egne innmeldinger	Høy	Rapportering	Ingen	Mulighet for å slette egen innmelding ved eventuelle feil.
Statusoppdatering	Middels	Rapportering	Ingen	Oppfølging til innmelder om hvor langt saken har kommet

Melde om events og hendelser i andre kategorier enn "feil og mangler"	Middels	Rapportering	Varierer fra kommune til kommune	Mulighet for enkelte ansatte i kommunen å bruke programmet til å informere om konserter o.l.
Lokal statistikk	Middels	Statistikk	Ingen	Oversikt over tidligere hendelser som er rapportert og løsningen som ble gjennomført ved de ulike sakene. Kommunal.
Nasjonal statistikk	Middels	Statistikk	Ingen	Oversikt over tidligere hendelser som er rapportert og løsningen som ble gjennomført ved de ulike sakene. Nasjonalt.
Kartreferanser	Høy	Input og database.	Ingen	Varsling må registreres med kartreferanser. Tenk over hvordan dette skal gjøres med hensyn til databasen. Muligheter for statistikk og lignende.

#### 4.5 Alternativer til vårt produkt

Det eksisterer noen lokale alternativer i ulike kommuner, men ingen nasjonale. Eksempler:

[Trondheim](#)

[Oslo](#)

[Bergen](#)

[Drammen](#)

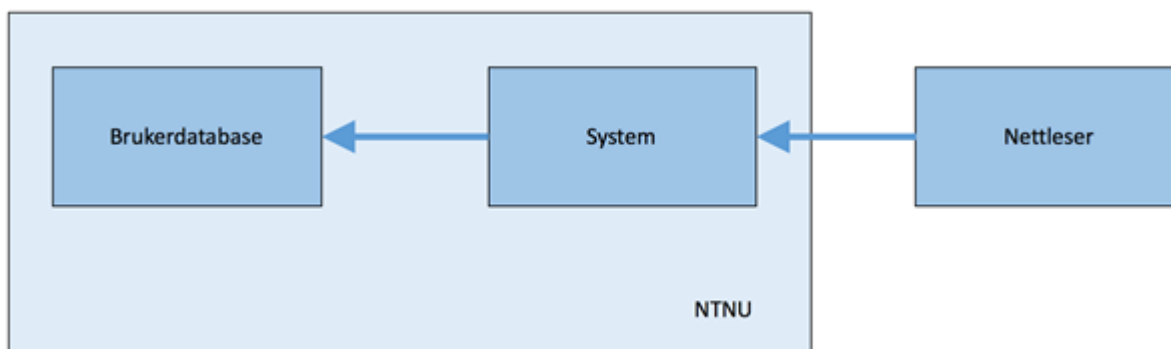
[Bodø](#)

[Vegvesen.no](#) er en informasjon-og varslingskanal for hele Norge, men det finnes ingen mulighet til å melde fra om feil og mangler på veisystemet eller annen infrastruktur.

## 5 Produktoversikt

### 5.1 Produktets rolle i brukermiljøet

Skisse av produktet med hensyn til omgivelsene, ta utgangspunkt i brukermiljøet (4.3)



### 5.2 Forutsetninger og avhengigheter

Registrering av bruker forutsetter at bruker har en e-postadresse. Alle som vil kan registrere seg selv som bruker i systemet.

## 6 Produktets funksjonelle egenskaper

En liste over funksjonelle egenskaper som produktet må ha for å løse problemene beskrevet i 4.4. NB! Det vil ikke nødvendigvis være et en-til-en forhold. Malen sier at mellom 25 og 99 funksjonelle egenskaper (funksjoner) er passe og at disse må være beskrevet såpass at de er et utgangspunkt for å utvikle dem ved hjelp av use-case.

Funksjonelle egenskaper
Funksjon for innlogging og oppretting av brukerkonto
Bruker kan registrere feil ved infrastruktur i sin egen kommune. Registrering av feil blir bekreftet ved en automatisert e-post som sendes til innmelder.
Funksjon for å distribuere og sette status på innmeldte feil.
Systemet må sjekke om feilen/mangelen allerede er rapportert

### 6.1 Alternativer til utvidelser

- Et poengsystem som gir varslere en digital valuta på samme måte som en rekke flyselskaper. Poengene/valutaen burde ha en reell verdi som skaper et insentiv, tenk kreativt.
- Utvidelse av appen utenfor kjernefunksjonalitet for eksempel legge til andre kommune-relevante funksjoner: parkering, turisme, kultur og lignende.



## 7 Ikke-funksjonelle egenskaper og andre krav

Det kreves at enhetstesting er implementert. På serversiden skal det være minst 50% dekningsgrad. På klienten er det ikke krav til dekningsgrad, men det er ønskelig med tester her også.

Alle data som benyttes av applikasjonen HverdagsHelt skal lagres i skolen MySQL-database. Databasefunksjonaliteten skal tilgjengeliggjøres for klienten via REST-tjenester.

En plattformuavhengig nettleser med støtte for en nyere HTML-standard må kunne brukes som klient mot applikasjonen.

Responstiden bør være under et sekund selv med 100 samtidige brukere. Det forventes en tilgjengelighet på 99.99% ("four nines" - tilsvarer maksimalt 52,56 minutter nedetid i året).

Løsningen skal ha god brukskvalitet, den skal være tilpasset målgruppen, lett og intuitivt og bruke for dem. Dette skal dokumenteres.

Løsningen skal ha god sikkerhet. Minstekrav er at løsningen sjekkes opp mot OWASP (2013) A1 og A3.

All kode skal lagres på enten GitLab eller GitHub.

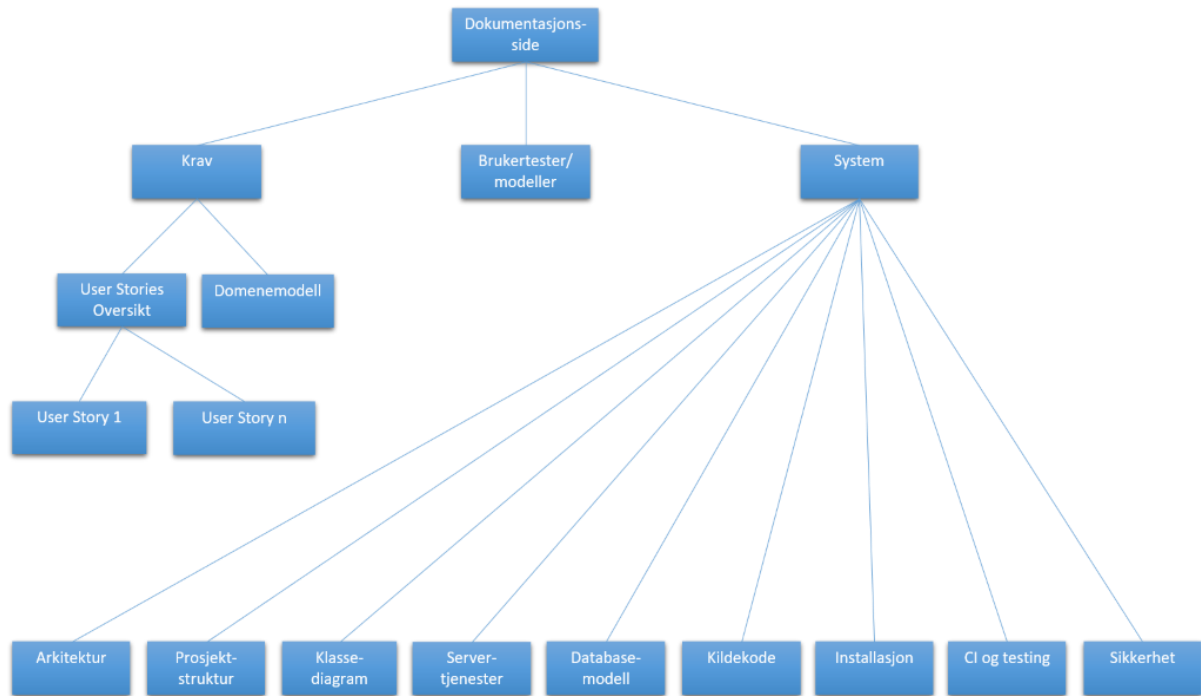
Sluttrapporten leveres som en PDF og skal inneholde følgende:

- Backlog/ Sprintbacklog sprint 1 og 2
- Sprintmål sprint 1 og 2
- Retrospektiv sprint 1 og 2
- Oppsummering av sprintene
- Refleksjonsnotat av hver deltaker i prosjektgruppen der en beskriver egen innsats i prosjektet og hvordan en opplevde samarbeidet i prosjektgruppen

Følgende dokumentasjon skal leveres som WIKI i GitLab eller GitHub

- Krav
- Systemdokumentasjon
- Brukertester/ konseptuelle modeller for brukergrensesnitt og lignende

## 7.1 Forslag til WIKI-struktur



### 7.1.1 Domenemodell

Lag en enkel domenemodell som beskriver problemdomenet.

### 7.1.2 User stories

Lag user stories som dekker all brukerfunksjonalitet for systemet. En user story kan skrives på denne formen:

Som <hvem?>  
Ønsker jeg <hva?>  
Slik at <hvorfor?>

Et eksempel kan være:

Som administrator  
Ønsker jeg å opprette brukerkontoer  
Slik at jeg kan gi nye brukere tilgang til systemet

Ta også med en kort liste med akseptansekriterier (krav) til hver user story. For eksempel:

- Gyldig epost, navn og telefonnummer må skrives inn
- Hvis epost er registrert fra før skal det vises en feilmelding.
- Systemet skal sende epost til den nye brukeren med et system-generert passord.
- Det systemgenererte passordet skal være på 12 tegn og bestå av helt tilfeldige tall eller små/store bokstaver.
- Passordene skal krypteres med «hashing med salt».

### **7.1.3 Brukertester/konseptuelle modeller**

Beskriv brukertester som er foretatt og eventuelle konseptuelle modeller for brukergrensesnitt som er brukt.

### **7.1.4 Arkitektur**

Lag en arkitekturskisse som beskriver de viktigste komponentene i systemet; klienter, server-tjenester, datakilder og kommunikasjonen mellom disse. Vis også eventuell lagdelt arkitektur for større komponenter.

En kort tekstlig beskrivelse bør være med for hver figur.

### **7.1.5 Prosjektstruktur**

Beskriv hvilke biblioteker og rammeverk dere bruker og vis fil- og katalogstrukturen for prosjektet; kildekode, pakker og hvordan disse er organisert.

### **7.1.6 Klassediagram**

Lag et overordnet klassediagram for serverdelen av løsningen. Hvis klasser ikke er i bruk kan dere levere en oversikt over moduler som er brukt.

Klassediagrammet skal også ha en tekstlig forklaring.

### **7.1.7 Server-tjenester**

Beskriv de ulike REST-ressursene. Dette gjelder også for eksempel WebSocket-endepunkter.

### **7.1.8 Databasemodell**

Vis en databasemodell her

Gi en kort forklaring av databasemodellen.

### **7.1.9 Dokumentasjon av kildekode**

Her kan dere beskrive og vise til dokumentasjon av kildekode. Ta med hvordan en eventuelt genererer og viser dokumentasjonen.

### **7.1.10 Installasjon og kjøring**

Lag installasjonsveiledning. Her må alt faglærer eller andre trenger å vite for å kunne installere og kjøre løsningen tas med.

### **7.1.11 Kontinuerlig integrasjon og testing**

Beskriv kontinuerlig integrasjon-oppsettet prosjektet benytter.

Beskriv også generelt testene som er laget; hvilke typer tester som er med, hva som er eller ikke er testet, og testdekning. Legg ved instruksjoner på hvordan en kjører testene.

### **7.1.12 Sikkerhet**

Beskriv kort hvordan sikkerheten i løsningen er ivaretatt