

Sluttrapport vedlegg
IDATT2106 team 10

Innhold

Innhold

| | |
|--|----|
| 1 Lenker til gitlab repo | 3 |
| 2 Lenke til GitLab wiki: | 3 |
| 3 Sprintkø for sprint 1 | 4 |
| 4 Sprintkø sprint 2 | 8 |
| 5 Gitlab wiki som pdf | 13 |
| Innhold | 13 |
| Funksjonelle krav | 14 |
| Ikke-funksjonelle krav | 20 |
| Undersider | 21 |
| Pipeline | 22 |
| Test | 23 |
| Test dekning | 23 |
| Klassediagram | 26 |
| 5.1 Brukertester | 46 |
| 6 WCAG-matrise | 53 |
| 7 Sprint planlegging referater | 55 |
| 8 sprint retrospektiv 1 referat | 56 |
| 9 sprint retrospektiv 2 referat | 58 |
| 10 sprint review 1 referat | 60 |
| 11 Sprint review 2 referat | 61 |
| 12 Møtereferat: 1. Scrum Standup | 63 |
| 13 Møtereferat: 2. Scrum Standup | 65 |
| 14 Møtereferat: 3. Scrum Standup | 68 |
| 15 Møtereferat: 4. Scrum Standup | 70 |
| 16 Møtereferat: 5. Scrum Standup | 70 |
| 17 Møtereferat: 6. Scrum Standup | 70 |
| 18 Møtereferat: 7. Scrum Standup | 70 |
| 19 Møtereferat: 8. Scrum Standup | 70 |
| 20 Møtereferat: 9. Scrum Standup | 70 |

| | |
|---|----|
| 21 Møtereferat: 10. Scrum Standup | 70 |
| 22 Møtereferat: 11. Scrum Standup | 70 |
| 23 Burndown chart sprint 1 | 71 |
| 24 burndown chart sprint 2 | 72 |
| 25 OWASP Dependency check | 73 |
| 26 brukerhistorier sprint 1 | 74 |
| 27 brukerhistorier sprint 2 | 75 |
| 28 ikke fullførte brukerhistorier | 76 |
| 29 Wireframes..... | 76 |
| | 77 |
| 30 Domenemodell..... | 78 |

1 Lenker til gitlab repo

Backend: https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/backend

Frontend: https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/frontend

2 Lenke til GitLab wiki:

https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/backend/-/wikis/home

3 Sprintkø for sprint 1









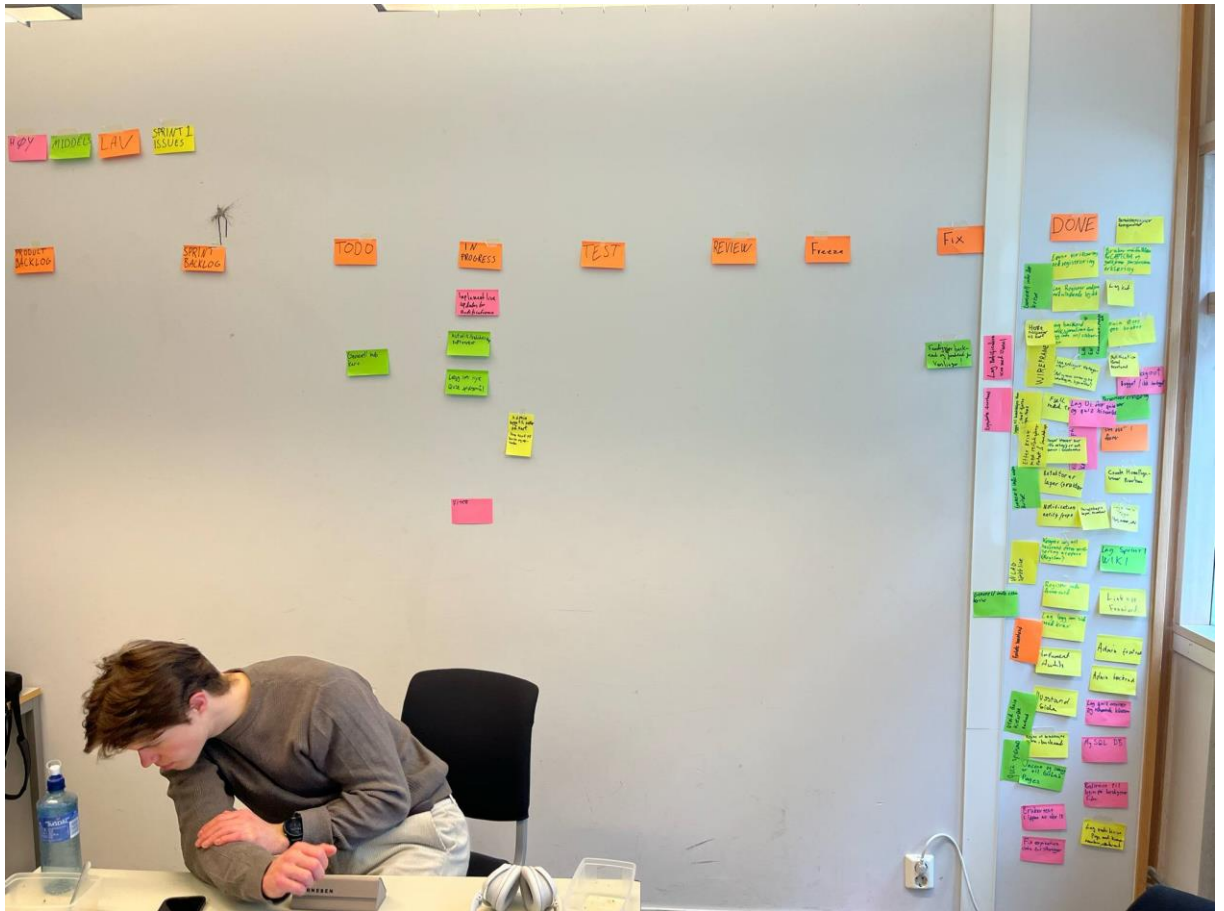
4 Sprintkø sprint 2













5 Gitlab wiki som pdf

title: Hjem

Innhold

- [Krav \(Home/Krav\)](#)
- [Brukertester \(Home/Brukertester\)](#)
- [System \(Home/System\)](#)
- [Bærekraftanalyse \(Home/B%C3%A6rekraftanalyse\)](#)
- [Brukertest 1 \(Home/Brukertester/Brukertest-1\)](#)
-

Bærekraftanalyse

Analyse av miljømessige, økonomiske og sosiale faktorer.

title: Krav

Funksjonelle krav

Tilfluktsrom på kart

Som ikke-registrert bruker ønsker jeg å kunne få opp mine nærmeste tilfluktsrom i et interaktivt kart slik at jeg

raskt kan få veibeskrivelse dit.

- Bruker må kunne se tilfluktsrom i et interaktivt kart
- Ikonet på kartet må være representativt og forklarende for funksjonaliteten

Se kriseberørt område

Som bruker ønsker jeg å ved inntreffelsen av en krise kunne se det berørte området på kartet, slik at jeg kan ha

opdatert informasjon om krisesituasjonen.

- Grad av alvorlighet skal kunne vises i fargekoder
- Brukeren skal kunne se sin posisjon på kartet, dersom de har på stedsdeling
- Kartet må oppdateres fortløpende når krisesituasjonen endrer seg
- Om brukeren berøres av krisen skal brukeren få et varsel (gjelder de som har på stedsdeling)

Registrere bruker

Som ikke-registrert bruker ønsker jeg å kunne registrere en ny brukerkonto og få tilgang til all funksjonalitet

tilgjengelig for registrerte brukere.

- Bruker må registrere seg ved bruk av en gyldig e-postadresse
- Bruker må kunne opprette eget passord som skal brukes ved innlogging
- Bruker må verifiseres med ReCAPTCHA før konto opprettes
- Bruker må bekrefte registrering via e-post med bekreftelseslenke
- Brukeren må godkjenne personvernerklæring for å fullføre registreringen

Bli med eller opprett husstand

Som ikke-registrert bruker ønsker jeg å kunne velge mellom å opprette ny husstand eller bli med i en eksisterende når jeg registrerer bruker, slik at jeg har valg om eget eller felles beredskapslager.

- Bruker må få spørsmål om å bli med eller opprette husstand ved registrering
- Bruker kan ikke bruke funksjoner for registrert bruker før valg er gjort

Generell informasjon (før/under/etter krise)

Som ikke-registrert bruker ønsker jeg å få generell informasjon og tips innen temaene “før krise”, “under krise”

og “etter krise”, slik at jeg kan få relevant informasjon knyttet til de ulike fasene.

- Bruker må kunne åpne informasjonssider for “før”, “under” og “etter” krise
- Brukeren skal ha tilgang til generelt, ikke-personalisert innhold på disse sidene

Personvern

Som ikke-registrert bruker ønsker jeg å ha tilgang til informasjon om personvern og hvordan data behandles for

alle brukere.

- Brukeren skal kunne åpne personvernsiden fra enhver side via en tilgjengelig lenke

Les nyheter

Som ikke-registrert bruker ønsker jeg å kunne lese nyhetsartikler, og se de nyeste øverst, slik at jeg holder meg

oppdatert.

- Bruker må kunne åpne nyhetsartikler
- Bruker må ha tilgang til både nye og eldre artikler
- Nyhetsseksjonen må vise nyeste artikler øverst sortert på publiseringsdato

Om oss

Som ikke-registrert bruker ønsker jeg informasjon om hvem som eier og drifter plattformen, samt kontaktinformasjon.

- Bruker må kunne se eier/drifter-info
- Bruker må kunne finne kontaktinformasjon til eier/driftsansvarlig

Logg inn

Som registrert bruker ønsker jeg å kunne logge inn med e-post og passord slik at jeg får tilgang til min beredskap og forblir pålogget.

- Passordfelt skal sensureres fortløpende
- Mulighet til å vise det sensurerte passordet
- Brukernavn-felt over passord-feltet
- Feilmelding ved feil passord eller e-post

Se beredskapslager

Som registrert bruker ønsker jeg å se husstandens beredskapslager inndelt i produktkategorier med antall/mengde og utløpsdato.

- Produkter delt inn etter produkttype
- Vise antall/mengde og utløpsdato for hvert produkt
- Lageret skal være felles for alle i husstanden

Oppdatere beredskapslager

Som registrert bruker ønsker jeg å legge til varer med mengde og utløpsdato, samt fjerne varer, slik at lageret

alltid er oppdatert.

- Knapp for "legg til ny vare" med felter for navn, mengde og utløpsdato
- Mulighet til å søke eksisterende produkt eller legge til nytt manuelt
- Endringer skal oppdateres for alle i husstanden

Antall i husstand

Som registrert bruker ønsker jeg å legge til/fjerne personer eller kjæledyr i husstanden, slik at beregning av

beredskapsgrad blir riktig.

- Ubegrenset antall personer/kjæledyr

Beredskapsgrad

Som registrert bruker ønsker jeg å se hvor lenge husstanden kan overleve basert på lageret, både før og under

krisen.

- Grafisk fremstilling av uker, dager og timer
- Beregnet matematisk ut fra registrert lager
- Oppdateres ved endringer i lager

Beredskapslager-kontrollvarsling

Som registrert bruker ønsker jeg varsler om snart utløpsdatoer, batteribytte osv., slik at lageret alltid er brukbart.

- Varsel én uke før utløpsdato
- Universell utforming på varsler (vibrasjon, lyd, skrift)
- Varslinger skal være korrekte i forhold til lagerdata

Møteplass på kart

Som registrert bruker ønsker jeg å se aktuelle møteplasser i et interaktivt kart slik at jeg raskt kan få veibeskrivelse dit.

- Bruker må ha én eller flere møteplasser definert
- Vise møteplasser i kart med forklarende ikoner

Hjemmeadresse på kart

Som registrert bruker ønsker jeg å se min hjemmeadresse som ikon i kartet slik at jeg kan vurdere nærhet til krisehendelser og POI.

- Mulighet til å fylle inn hjemmeadresse i brukerinformasjon
- Visualisere adresse som interaktivt kartikon

Invitere til husstand

Som registrert bruker ønsker jeg å kunne invitere andre til min husstand slik at vi deler samme beredskapslager og varsler.

- Deling av lager og varsler skal være likt for alle i husstanden

Opprette beredskapsgruppe

Som registrert bruker ønsker jeg å opprette en gruppe slik at jeg kan invitere andre husstander i nærområdet til å dele forsyninger.

- Bruker må kunne navngi beredskapsgruppen

Legge til beredskap i gruppe

Som registrert bruker ønsker jeg å velge hvilke varer fra mitt lager som skal deles i gruppen, slik at jeg ikke deler

alt om jeg ikke vil.

- Velge varer og mengder fra eget lager
- Variable delingsinnstillinger for gruppe-medlemmer

Delt beredskapslager

Som registrert bruker ønsker jeg å se status på gruppens lager, slik at jeg vet hva og hvor mye hver husstand

har.

- Vise hvilke varer, totale mengder og utløpsdatoer per husstand
- Egen oversikt separat fra eget husstandslager

Se posisjon til andre

Som registrert bruker ønsker jeg å se sanntidsposisjon til husstands- og gruppe-medlemmer som har aktivert

stedsdeling.

- Kun vise posisjon til brukere som har akseptert stedsdeling
- Alle vises på samme kart med identifisering

Glemt passord

Som registrert bruker ønsker jeg å kunne tilbakestille eller bytte passord ved glemt passord.

- Send e-post med tilbakestillingslenke
- Nytt passord skal virke umiddelbart

Refleksjon etter krise

Som registrert bruker ønsker jeg å skrive fritekstrefleksjoner etter en krise, slik at jeg kan evaluere og justere

beredskap.

- Skrive inn fritekst
- Dele refleksjoner med andre brukere
- Se historikk over tidligere notater

Bytte husstand

Som registrert bruker ønsker jeg å bytte husstand jeg tilhører, slik at jeg kan administrere et annet beredskapslager.

- Tildeling og fjerning via invitasjon
- Valg av hvilke varer medfølger til ny husstand; fjerner dem fra gammel
- En brukerløs husstand slettes og gjenværende varer slettes

Varsling om stedsdeling

Som registrert bruker ønsker jeg å få varsel om å aktivere stedsdeling ved krise, slik at jeg kan bevare personvern før event. krise.

- Mulighet til å aktivere/deaktivere stedsdeling under og etter
- registrering Ved krise: varsel med enkel knapp for å aktivere stedsdeling raskt

Gamification

Som registrert bruker ønsker jeg engasjerende læringsaktiviteter gjennom gamification, slik at jeg lettere lærer

om beredskap.

- Se historikk over tidligere aktiviteter

Admin-innlogging

Som admin ønsker jeg et sikkert passord og totrinns autentisering for admin-konto, slik at jeg kan administrere

løsningen.

- Opprette adminbruker via link fra superadmin
- Passordkrav: min. 8 tegn, store/små bokstaver, tall, spesialtegn
- To-trinns autentisering ved hver innlogging
- Tilbakestillingslenke gyldig 1 time
- Brukerkrav mot brute-force-angrep

Kart-ikoner (admin)

Som admin vil jeg kunne opprette, redigere og slette posisjoner (hertestarter, tilfluktsrom, matstasjon osv.) på

kartet.

- Legg til ved klikk på kart eller adresse/koordinater
- Velg type posisjon, legg til beskrivelse, åpningstider, kontakinfo
- Umiddelbar visning på kartet
- Kun admin-tilgang, søk- og filterfunksjonalitet
- Feilmeldinger ved manglende data

Kartoppdateringer (hendelser)

Som admin vil jeg kunne opprette, oppdatere og slette krisehendelser slik at brukere får sanntidsinformasjon.

- Angi tittel, beskrivelse, krisenivå, tidsstempel, posisjon
- Umiddelbar visning og oppdatering i UI
- Slette hendelser slik at de ikke lenger vises
- Varsle brukere i nærheten ved nye endringer

Scenarier (admin)

Som admin vil jeg kunne opprette og vedlikeholde krisescenarier slik at brukere lett får informasjon om ulike

typer kriser.

- Opprette scenario-tema (f.eks. strømbrytning)
- Redigere faktside innen hvert scenario

Gamification-admin

Som admin vil jeg kunne opprette, oppdatere og slette gamification-aktiviteter slik at brukere lærer om krise og

beredskap.

- Opprette nye aktiviteter, angi navn og emne/gruppe
- Redigere og slette eksisterende aktiviteter

Invitere superadmin

Som superadmin vil jeg kunne sende invitasjonslenke til e-post slik at nye admin-brukere kan legges til.

- Legge til og fjerne admin
- Sende "nytt passord"-lenke til admin

Ikke-funksjonelle krav

- Det kreves programmatisk testing av koden:
 - Serverside: minst 50 % dekningsgrad (enhet, integrasjon, E2E)
 - Klient: minst 30 % dekningsgrad, fortrinnsvis enhetstester
- All data skal lagres i skolens MySQL-database (eller tilsvarende oppspunnet DB som initieres ved oppstart),
- Data skal tilgjengeliggjøres via REST-tjenester
-

Klienten skal være plattformuavhengig i nettleser med støtte for nyere HTML-standard; minimum Chrome og Firefox

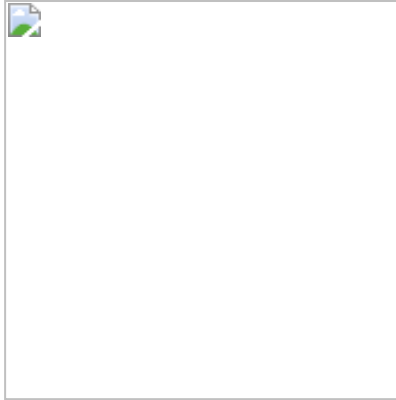
- Løsningen skal være i samsvar med WCAG 2.1 prinsipp 1 (Begripelig) – god brukskvalitet, målgruppetilpasset, intuitiv, og dokumentert
 - Løsningen skal ha god sikkerhet:
 - Autentisering og autorisering implementert
 - Sjekkes mot OWASP A03:2021-Injection (inkl. XSS)
 - All kode skal versjoneres på skolens GitLab under prosjektnavn `idatt2106_2025_teamnr`; faglærere skal ha Reporter-tilgang; wiki-landingsside må inkludere teamnr og fullt navn på alle medlemmer
 - Bruk av byggesystemer som kjører tester og gjør det enkelt å bygge og kjøre applikasjonen
 - Serverside skal være REST-API, og **Spring Boot** er påkrevd
 - Klient skal være en SPA; rammeverk fritt valg (annet enn Vue tillatt ved enighet)
 - Continuous Integration (CI) med testrapporter i GitLab er påkrevd og Continuous Delivery (CD) er valgfritt
 - [Sprint 1 \(Home/Sprint-Reviews/Sprint-1\)](#)
 - [Sprint 2 \(Home/Sprint-Reviews/Sprint-2\)](#)
-

title: System

Oversikt over systemdesign og implementasjon.

Undersider

- [Arkitektur \(Home/System/Arkitektur\)](#)
- [Prosjektstruktur \(Home/System/Prosjektstruktur\)](#)
- [Klassediagram \(Home/System/Klassediagram\)](#)
- [Servertjenester \(Home/System/Servertjenester\)](#)
- [Databasemodell \(Home/System/Databasemodell\)](#)
- [Kildekode \(Home/System/Kildekode\)](#)
- [Installasjon \(Home/System/Installasjon\)](#)
- [CI og Testing \(Home/System/CI-og-Testing\)](#)
- [Sikkerhet \(Home/System/Sikkerhet\)](#)



- **Klient**

Web- eller mobilklient som sender HTTP-forespørsler og mottar JSON-svar.

- **Security Filter Chain (Spring Security)**

Autentiserer og autoriserer alle innkommende forespørsler før de når controllerne.

- **Rest Controller**

Eksporerer API-endepunkter, konverterer JSON til Request DTO og gir Response DTO tilbake.

- **Global Exception Handler**

Fanger opp unntak fra controller/service og returnerer konsistente HTTP-feilmeldinger.

- **Service**

Inneholder forretningslogikk, validering og orkestrering av kall til repository-laget.

- **Repository**

JPA-grensesnitt som utfører CRUD-operasjoner og egendefinerte spørringer mot databasen.

- **MySQL Database**

Vedvarende lagring av Entity-objekter via Hibernate/JPA.

- **Eksterne datakilder**

- **Kartverket API** for geokoding og adresser
- **GeoJSON**-filer med tilfluktsrom lokalisert i systemet!![Systemarchitecure.drawio]

Applikasjonen blir ikke deployet men det kjøres enn pipeline hvergang noe pushes til repositoriet.

Pipeline

Hvilke steg som kjøres i pipeline er avhengig av hvilken branch som blir pushet til. Det er tre forskjellige brancher som

har forskjellige steg.

Felles pipeline steg

- **Build:** På backend bygges applikasjonen med 'mvn clean compile' og frontend bygges med 'npm run build'.

Test: Backend kjører unit tester med 'mvn test' og frontend kjører unit tester med 'npm run test'.

Pages backend

Når det pushes til branchen 'dev' på backend kjøres først felles stegene. Deretter kjøres:

- **generate-openapi:** Genererer openapi spesifikasjonen fra backend og lagrer den i 'openapi.json' i root mappen.
- **deploy:** Deployer jaccoco test rapporten til github pages. På backend deployes også swagger dokumentasjonen til github pages.

Pages frontend

Når det pushes til branchen 'dev' på frontend kjøres først felles stegene. Deretter kjøres:

- **deploy:** Deployer istanbull test rapporten til github pages.

Security

På frontend når det pushes til branchen 'chore/owasp-check-integration' kjøres først steget

- **dependency-check-job:** Kjører OWASP dependency check for å sjekke om det er kjente sårbarheter i avhengighetene.
- Deretter kjøres felles stegene.

Test

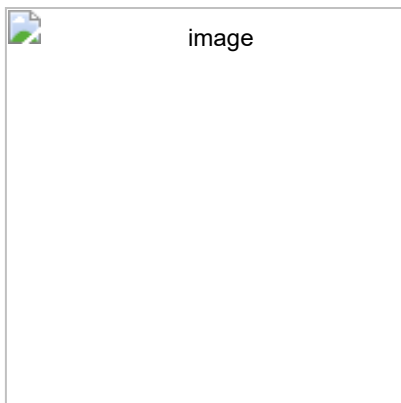
Backend har både enhetstester og integrasjonstester som kjøres i pipeline. Frontend har enhetstester som kjøres i

pipeline. Det er også brukt end-to-end tester med Cypress. Cypress testene kjøres ikke i pipeline.

Dersom noen av pipeline stegene feiler får man ikke lov å merge til dev branchen og det deployes ikke til github pages.

Test dekning

- Backend har krav om 50% test dekning og har totalt 57% test dekning.
- Frontend har krav om 30% test dekning og har 30% dekning på unit tester i tillegg til å ha end-to-end tester med Cypress.



Denne databasen er bygd opp rundt fem

hovedområder: 1. Bruker- og rollehåndtering

- *users*, *user_roles* og *verification_tokens* styrer registrering, autentisering og tilgangsnivå for hver bruker. Brukere kan tilhøre en *household* (husstand). Og husstander kan ha invitasjonstokens (*household_token*) til å invitere nye medlemmer til sin husstand.

2. Quiz-modul

- Spørsmål (*questions*) har flere svaralternativer (*answer_options*). Hver gang en bruker tar en quiz, opprettes en *quiz_attempt* med tilhørende *quiz_attempt_answers* som kobler hver besvart *question* til et valgt *answer_option*.

3. Lager og varsler for husstand

- *item_definitions* og *item_category* beskriver hva slags varer som finnes. Hver husstand har egne *storage_item*-oppføringer som registrerer beholdning og sist oppdatering. På bakgrunn av lagerstatus genereres *notifications* (f.eks. lav beholdning eller utløpte varer).

4. Geodata & hendelser

- *event* (med type fra *event_type*) og *position* (med type fra *position_type*) lagrer henholdsvis dynamiske karthendelser og faste lokasjoner med geometri, tidsstempel og status.

5. Refleksjonsnotater

- *reflection_notes* lar brukerne skrive og lagre egne notater, private eller delte.

Denne siden beskriver hvordan du raskt setter opp **Krisefikser** for lokal utvikling og testing. Her finnes kun det som er

nødvendig for å komme i gang.

1. Forutsetninger

- **Backend**

- Java JDK 21
- Maven
- MySQL

- **Frontend**

- npm

Merk: Du trenger `.env` filer for både frontend og backend, få disse fra en av prosjektetgruppens medlemmer før du starter.

2. Backend

2.1 Kloner repository

```
git clone
https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/backend.git cd
backend
```

2.2 Kjør i utviklingsmodus

Starter Spring Boot mot lokal MySQL:

```
mvn spring-boot:run \
-Dspring-boot.run.profiles=dev
```

- Applikasjonen kjører på `https://localhost:8080` #TODO

2.3 Kjør med in-memory DB for testing

```
mvn spring-boot:run \
-Dspring-boot.run.profiles=test
```

2.3 Kjør tester

```
mvn test
```

3. Frontend

3.1 Kloner repository

```
git clone
https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/frontend.git cd
frontend
```

3.2 Installer avhengigheter

```
npm install
```

3.3 Start utviklingsserver

```
npm run dev
```

- Frontend er tilgjengelig på <http://localhost:5173>

Etter disse stegene har du både backend og frontend kjørende lokalt, klare for utvikling og testing.

Kildekoden for backend finnes her: https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/backend
(https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/backend)

Og kildekoden for frontend finnes her: https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/frontend
(https://gitlab.stud.idi.ntnu.no/idatt2106_2025_team10/frontend)

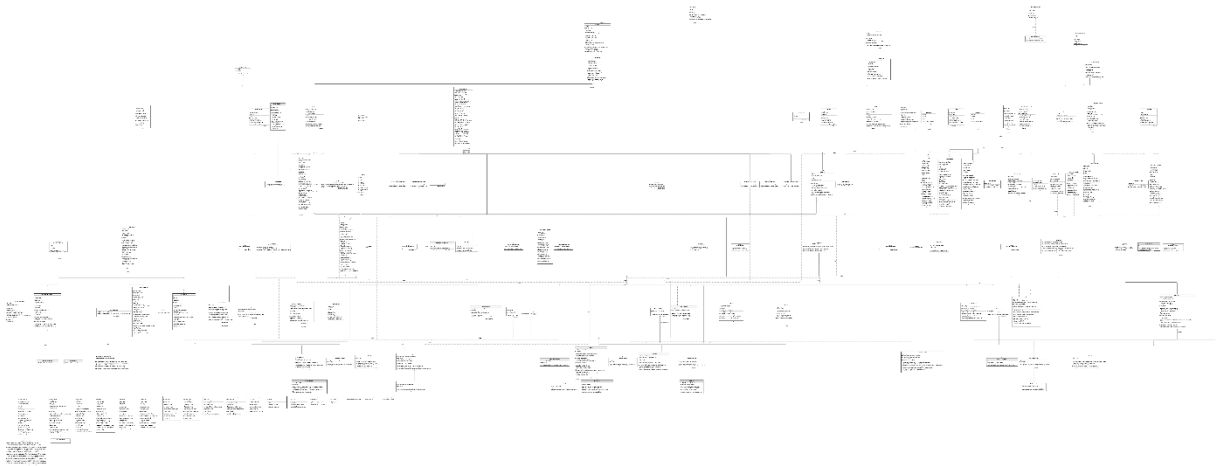
All javakoden er dokumentert med javadoc, og swagger. Swagger dokumentasjonen finnes ved:
[https://backend-](https://backend-91339c.pages.stud.idi.ntnu.no/swagger-ui/)

[91339c.pages.stud.idi.ntnu.no/swagger-ui/](https://backend-91339c.pages.stud.idi.ntnu.no/swagger-ui/) (<https://backend-91339c.pages.stud.idi.ntnu.no/swagger-ui/>) og
info om

endepunktene finnes også [her](#) (Home/System/Servertjenester)

Klassediagram

UML-diagrammer for klassestrukturen.



Biblioteker og rammeverk

Backend

- **Java 21** - JDK
- **Maven** - Byggeverktøy
- **Spring Boot Starter** - kjerne-Spring Boot-rammeverk
- **Spring Boot Starter Web** - REST-API
- **Spring Boot Starter Security** - autentisering/autorisation
- **Spring Security Test** - teststøtte for sikkerhet
- **Spring Boot Starter Mail** - e-post (f.eks. verifisering)
- **Spring Boot DevTools** - hot reload under utvikling
- **Spring Boot Starter Data JPA** - ORM med Hibernate
- **Spring Boot Starter WebSocket** - WebSocket
- **Springdoc OpenAPI Starter WebMVC UI** - Swagger - dokumentasjon
- **Hibernate Spatial** - romlig databehandling
- **JTS Core & IO-Common** - geometri-
- **API H2GIS** - H2-database med GIS-
- utvidelse
- **Proj4J** - koordinatprojeksjoner
- **MySQL Connector/J** - MySQL JDBC-driver
- **H2 Database** - in-memory database for utvikling og tester
- **JWT** - JSON Web Tokens
- **Log4j2** - logging
- **Spring Boot Starter Test** - JUnit 5, Mockito, AssertJ, etc.

Frontend

- **Vue 3**
 - **TypeScript**
 - **Vite** – dev-server og bundling
 - **Tailwind CSS** – CSS-
 - rammeverk **Pinia** –
 - tilstandshåndtering for Vue
 - **vue-router** – klient-side routing
 - for Vue **Axios** – HTTP-klient
 - for API-kall
 - **Leaflet** – interaktivt kartbibliotek **OL**
 - (**OpenLayers**) – kart-API **ol-cesium** –
 - integrasjon av Cesium med OpenLayers **ol-**
 - **mapbox-style** – Mapbox-styler i OpenLayers
 - **olcs** – bro mellom OpenLayers og Cesium
 - **Proj4** – koordinatprosjeksjoner
 - **SockJS-client** – fallback WebSocket-klient
 - **@stomp/stompjs** – STOMP-protokoll over WebSocket
 - **Vitest** – testrammeverk for enhetstester
 - **Cypress** – end-to-end-testing
 - **@headlessui/vue** – headless UI-
 - komponenter **@heroicons/vue** – ikoner
 - tilpasset Vue **headlessui** – generell
 - Headless UI-pakke **heroicons** – generell
- Heroicons-pakke
- @storefront-ui/vue** – UI-komponentbibliotek for Vue
- Sentry** – feilrapportering og ytelsessporing
- **Turf.js** – geospasiale funksjoner
 - **@vueuse/core** – samling av nyttige Vue-composables
 - **@vuepic/vue-datepicker** – datovelger-
 - komponent **Cesium** – 3D kartmotor **class-**
 - **variance-authority** – CSS-klasse-
 - variantgenerator **clsx** – enkel sammenslåing av
 - class-strings **reka-ui** – UI-komponentbibliotek **oh-**
 - **vue-icons** – ikonpakke for Vue **lucide-vue-next**
 - – Lucide-ikoner for Vue Next **Sass** – CSS-
 - preprocessor **lodash.debounce** – debounce-
 - funksjon **vue-multiselect** – multiselect-
 - komponent for Vue
 - **Vuetify 3** – Material Design-komponentrammeverk for Vue

Prosjektstruktur

Programmkoden i prosjektet er delt inn i 2 repositories, en for backend og en for frontend.

Backend:


```
└─ backend
  └─ src
    └─ main
      └─ java
        └─ lombok.config
        └─ stud
          └─ ntnu
            └─ no
              └─ krisefikser
                └─ config
                  └─ JWTAuthorizationFilter.java
                  └─ JWTUtil.java
                  └─ SecurityConfig.java
                  └─ SecurityUtil.java
                  └─ SwaggerConfig.java
                  └─ websocket
                    └─ JwtHandshakeInterceptor.java

└─ WebSocketConfig.java

└─ controller
  └─ admin
    └─ AuthController.java
  └─ AuthController.java
  └─ HouseholdController.java
  └─ ItemCategoryController.java
  └─ map
    └─ EventController.java
    └─ EventTypeController.java
    └─ PositionController.java
    └─ PositionTypeController.java
  └─ NotificationController.java
  └─ PreparednessController.java
  └─ QuizController.java
  └─ ReflectionController.java
  └─ StorageItemController.java

└─ UserController.java

└─ dtos
  └─ auth
    └─ LoginRequest.java
    └─ RegisterRequest.java
  └─ event
  └─ household
    └─ HouseholdRequest.java
    └─ HouseholdResponse.java
```

```

└─ itemCategory
    └─ ItemCategoryRequest.java
    └─ ItemCategoryResponse.java
└─ map
    └─ event
        └─

CircleDataDto.java
└─ EventRequest.java

└─ EventResponse.java
└─ EventTypeRequest.java
└─ EventTypeResponse.java
└─ position
    └─ PositionRequest.java
    └─ PositionResponse.java
    └─ PositionTypeRequest.java
    └─ PositionTypeResponse.java
└─ shelters
    └─ ShelterRequest.java
    └─ ShelterResponse.java
└─ mappers
    └─ CircleDataMapper.java
    └─ EventMapper.java
    └─ EventTypeMapper.java
    └─ HouseholdMapper.java
    └─ ItemCategoryMapper.java
    └─

NotificationMapper.java
└─ PositionMapper.java

└─ PositionTypeMapper.java
└─ QuizMapper.java
└─ reflections
    └─ ReflectionNoteMapper.java
└─ StorageItemMapper.java
└─ UserMapper.java
└─ preparedness
    └─ NotificationRequest.java
    └─ NotificationResponse.java
    └─ PreparednessResponse.java
    └─ PreparednessSummary.java
└─ quiz
    └─ QuizAnswerOptionDto.java
    └─ QuizAnswerRequest.java

```


- └ QuizAnswerResponse.java
 - └ QuizAttemptHistoryDto.java
 - └ QuizAttemptQuestionDto.java
 - └ QuizAttemptResponse.java
 - └ QuizQuestionDto.java
 - └ QuizRequest.java
 - └ QuizResponse.java
 - └ QuizResultResponse.java
- └ reflection
 - └ ReflectionNoteRequest.java
 - └ ReflectionNoteResponse.java
- └ shelters
- └ storageItem
 - └ StorageItemResponse.java
 - └
- StorageItemStockUpdateRequest.java
 - └ user
 - └ UserResponse.java
- └ entities
 - └ Household.java
 - └
- HouseholdToken.java
 - └ ItemCategory.java
- └ ItemDefinition.java
- └ map
 - └ CircleData.java
 - └ Event.java
 - └ EventSeverity.java
 - └ EventStatus.java
 - └ EventType.java
 - └ Position.java
 - └
- PositionType.java
 - └
- Notification.java
 - └ NotificationType.java
 - └ QuizAnswerOption.java
 - └ QuizAttempt.java
 - └
- QuizAttemptAnswer.java
 - └ QuizQuestion.java
- └ QuizSessionStatus.java
- └ reflections
 - └ ReflectionNote.java

- └─ ReflectionNoteVisibility.java
- └─ Role.java
- └─ StorageItem.java
- └─ User.java
- └─ UserRole.java
- └─ VerificationToken.java
- └─ exception
 - └─ CustomErrorMessage.java
 - └─ customExceptions
 - └─ AppEntityNotFoundException.java
 - └─
- EditedValueUnchangedException.java
- └─
- EntityAlreadyExistsException.java
 - └─ EntityOperationException.java
 - └─ InvalidGeoJsonException.java
 - └─
- NotificationNotFoundException.java
- └─ UnauthorizedOperationException.java
 - └─ ErrorDetail.java
 - └─ GlobalExceptionHandler.java
- └─ KrisefikserApplication.java
- └─ logging
 - └─ LoggingAspect.java
- └─ repository
 - └─ EventRepository.java
 - └─ EventTypeRepository.java
 - └─ HouseholdRepository.java
 - └─
- HouseholdTokenRepository.java
- └─ ItemCategoryRepository.java
 - └─ ItemDefinitionRepository.java
 - └─
- NotificationRepository.java
- └─ PositionRepository.java
 - └─ PositionTypeRepository.java
 - └─ QuizAnswerOptionRepository.java
 - └─ QuizAttemptAnswerRepository.java
 - └─ QuizAttemptRepository.java
 - └─ QuizQuestionRepository.java
 - └─ ReflectionRepository.java
 - └─ StorageItemRepository.java
 - └─ UserRepository.java

```
└─ VerificationTokenRepository.java
└─ service
    └─ AdminService.java
    └─ CaptchaService.java
    └─ CustomUserDetailsService.java
    └─ EventService.java
    └─ EventTypeService.java
    └─ HouseholdService.java
    └─ ItemCategoryService.java
    └─
NotificationService.java
└─ PositionService.java
    └─ PositionTypeService.java
    └─ PreparednessService.java
    └─ QuizService.java
    └─ ReflectionService.java
```

```
└─ ShelterService.java
└─ StorageItemCreationService.java
└─ StorageItemService.java
└─ UserService.java
└─ util
    └─ DateUtil.java
└─ resources
    └─ application-dev.properties
    └─ application-test.properties
    └─ application.properties
    └─ log4j2.xml
    └─ tilfluktsrom.geojson
└─ test
    └─ java
        └─ stud
            └─ ntnu
                └─ no
                    └─ krisefikser
                        └─ controller
                            └─ AuthControllerIntegrationTest.java
                            └─ map
                                └─ QuizControllerIntegrationTest.java
                        └─ KrisefikserApplicationTests.java
                    └─ service
                        └─ EventServiceTest.java
                        └─ HouseholdServiceTest.java
                        └─ ItemCategoryServiceTest.java
                        └─ NotificationServiceTest.java
                        └─ PreparednessServiceTest.java
                        └─ QuizServiceTest.java
                        └─ ReflectionServiceTest.java
                        └─ ShelterServiceTest.java
                        └─ StorageItemServiceTest.java
└─ UserServiceTest.java
└─ .env
└─ .gitignore
└─ .gitlab-ci.yml
└─ mvnw
└─ mvnw.cmd
└─ pom.xml
└─ README.md
```

Frontend:

```
└─ frontend
  └─ cypress
    └─ e2e
      └─ example.cy.ts
    └─ fixtures
      └─ example.json
    └─ support
      └─ commands.ts
      └─ e2e.ts
    └─ tsconfig.json
  └─ public
    └─ icons
      └─ android-chrome-192x192.png
      └─ android-chrome-512x512.png
      └─ apple-touch-icon.png
      └─ favicon-16x16.png
      └─ favicon-32x32.png
      └─ favicon.ico
      └─ map
        └─ custom-shelter.svg
  └─ food-station.svg
    └─ health-station.svg
    └─ house-svgrepo-com.svg
    └─ shelter-icon.svg
    └─ user-location.svg
    └─ site.webmanifest
  └─ src
    └─ __tests__
      └─ after-crisis
        └─ InfoCard.test.ts
        └─ PublicOverview.test.ts
      └─ auth
        └─ Login.test.ts
        └─ Register.test.ts
        └─ VerificationSent.test.ts
        └─ Verify.test.ts
      └─ before-crisis
        └─ BeforeCrisisDashboard.test.ts
  └─ BeforeCrisisUnderPage.test.ts
    └─ Container.test.ts
    └─ crisis
      └─ UnderDashboard.test.ts
      └─ UnderPage.test.ts
```

- └─ ErrorMessage.test.ts
 - └─ footer
 - └─ Footer.test.ts
 - └─
- AboutUs.test.ts
- └─ Footer.test.ts
 - └─ Privacy.test.ts
- └─ home
 - └─ HomePage.test.ts
 - └─ MainDashboard.test.ts
 - └─ InviteToHousehold.test.ts
 - └─ JoinHousehold.test.ts
 - └─ LeaveHousehold.test.ts
 - └─ map
 - └─ Map.test.ts
 - └─ MapAdminPanel.test.ts
 - └─ MapUserPanel.test.ts
 - └─ MapView.test.ts
 - └─ NavBar.test.ts
 - └─ PreparationGroup.test.ts
 - └─ reflection
 - └─ CreateReflectionNoteModal.test.ts
 - └─ ReflectionNoteCard.test.ts
- └─ ReflectionNoteDetailModal.test.ts
 - └─ storage
 - └─ EmergencyStorageDashboard.test.ts
 - └─ EmergencyStorageGrid.test.ts
 - └─ EmergencyStorageItem.test.ts
 - └─ NotificationPanel.test.ts
 - └─ PreparednessSummaryCard.test.ts
- └─ assets
 - └─ before-crisis.svg
 - └─ logoSys.png
 - └─ main.css
 - └─ under-crisis.svg
- └─ components
 - └─ __tests__
 - └─ AddressSelector.vue
 - └─ admin
 - └─ AdminDashboard.vue
 - └─ AdminLogin.vue
 - └─ after-crisis

- └─
- AfterDashboard.vue
 - └─ InfoCard.vue
 - └─ PublicOverview.vue
 - └─ auth
 - └─ Login.vue
 - └─ Register.vue
 - └─ VerificationSent.vue
 - └─ Verify.vue
 - └─ before-crisis
 - └─ BeforeCrisisDashboard.vue
 - └─ BeforeCrisisUnderPage.vue
 - └─ Container.vue
 - └─ crisis
 - └─ UnderDashboard.vue
 - └─ UnderPage.vue
 - └─ ErrorMessage.vue
 - └─ footer
 - └─
- AboutUs.vue
 - └─ Footer.vue
 - └─ Privacy.vue
 - └─ home
 - └─
- DashboardCard.vue
 - └─ HomePage.vue
 - └─ MainDashboard.vue
 - └─ icons
 - └─ InviteToHousehold.vue
 - └─ JoinHousehold.vue
 - └─ LeaveHousehold.vue
 - └─ map
 - └─ controls
 - └─ Map.vue
 - └─ MapAdminPanel.vue
 - └─ MapUserPanel.vue
 - └─ MapView.vue
 - └─ NavBar.vue
 - └─ PreparationContainer.vue
 - └─ PreparationGroup.vue
 - └─ reflection
 - └─ CreateReflectionNoteModal.vue


```
└─ ReflectionNoteCard.vue
└─ ReflectionNoteDetailModal.vue
    └─ storage
        └─ EmergencyStorageDashboard.vue
        └─ EmergencyStorageGrid.vue
        └─ EmergencyStorageItem.vue
        └─ NotificationPanel.vue
        └─ PreparednessSummaryCard.vue └─ composables
    └─ useApi.ts
    └─ useGeoLocation.ts
    └─ useGlobalEditing.ts
    └─ useMap.ts
└─ KriseFikser.vue
└─ lib
    └─ utils.ts
└─ main.ts
└─ router
    └─ index.ts
└─ services
    └─ addressApi.ts
    └─ adminApi.ts
    └─ apiClient.ts
    └─ authApi.ts
    └─ EmergencyStorageApi.ts
    └─ householdApi.ts
    └─ kartverketClient.ts
    └─ mapApi.ts
    └─ quizApi.ts
    └─ reflectionApi.ts
    └─ userApi.ts
└─ stores
    └─ household.ts
    └─ mapStore.ts
    └─ notification.ts
    └─ useLocationStore.ts
    └─ user.ts
└─ types
    └─ AuthDto.ts
    └─ leaflet-pm.d.ts
    └─ sockjs-client.d.ts
    └─ types.ts
    └─ UserDto.ts
```

```
└─ utils
  └─ handleApiError.ts
  └─ mapUtils.ts
  └─ socket.ts
└─ views
  └─ admin
    └─ AdminDashboardView.vue
    └─ AdminLoginView.vue
  └─ AfterCrisisView.vue
  └─ auth
    └─ LoginView.vue
    └─ RegisterView.vue
  └─ BeforeCrisisView.vue
  └─ EmergencyStorageView.vue
  └─ HomePageView.vue
  └─ HouseholdView.vue
  └─ MyReflectionsView.vue
  └─ QuizView.vue
  └─ ReflectionNotesView.vue
  └─ UnderCrisisView.vue
  └─ VerifyEmailView.vue
└─ .env
└─ .gitignore
└─ .gitlab-ci.yml
└─ components.json
└─ cypress.config.ts
└─ env.d.ts
└─ eslint.config.ts
└─ index.html
└─ package-lock.json
└─ package.json
└─ README.md
└─ tailwind.config.ts
└─ tsconfig.app.json
└─ tsconfig.json
└─ tsconfig.node.json
└─ tsconfig.vitest.json
└─ vite.config.ts
└─ vitest.config.ts
```

REST-endepunkter

Autentisering

- `POST /api/auth/register` – Registrer ny bruker
- `POST /api/auth/login` – Logg inn
- `POST /api/auth/logout` – Logg ut
- `GET /api/auth/verify/{token}` – Verifiser e-
- `post GET /api/auth/me` – Hent profil for innlogget bruker

Husstand

- `POST /api/household/create` – Opprett ny husstand
- `POST /api/household/invite` – Inviter bruker til husstand
- `POST /api/household/join/{householdToken}` – Bli med i eksisterende husstand
- `POST /api/household/leave` – Forlat husstand
- `GET /api/household/my` – Hent innlogget brukers husstand

Brukerprofil

- `GET /api/users/profile` – Hent profil for innlogget bruker

Varekategorier

- `GET /api/itemcategories` – Hent alle
- `varekategorier POST /api/itemcategories` – Opprett ny varekategori

Lagerartikler

- `PUT /api/storageitems/update-stock` – Oppdater beholdning for en lagerartikkel
- `GET /api/storageitems/{householdId}` – Hent alle lagerartikler for en husstand
- `GET /api/storageitems/{householdId}/{categoryId}` – Hent lagerartikler for en husstand filtrert på kategori

Beredskap

- `GET /api/preparedness/{householdId}` – Hent beredskapsoversikt for husstand

Varslinger

- `POST /api/notifications` – Opprett varsel
- `GET /api/notifications/household/{householdId}` – Hent varsler for husstand
- `DELETE /api/notifications/{id}` – Slett varsel

Hendelsestyper

- `GET /api/event-types` – Hent alle hendelsestyper
- `GET /api/event-types/{id}` – Hent hendelsestype etter ID

Posisjonstyper

- `GET /api/position-types` – Hent alle posisjonstyper

Posisjoner

- `GET /api/positions` – Hent alle posisjoner
- `GET /api/positions/{id}` – Hent posisjon etter ID
- `PUT /api/positions/{id}` – Oppdater eksisterende posisjon
- `DELETE /api/positions/{id}` – Slett posisjon
- `POST /api/positions` – Legg til ny posisjon
- `POST /api/positions/shelters-nearby` – Finn tilfluktsrom innen gitt radius
- `GET /api/positions/shelters` – Hent alle tilfluktsrom

Hendelser

- `GET /api/events` – Hent alle hendelser
- `GET /api/events/{id}` – Hent hendelse etter ID
- `PUT /api/events/{id}` – Oppdater hendelse
- `DELETE /api/events/{id}` – Slett hendelse
- `POST /api/events` – Opprett ny hendelse
- `GET /api/events/eventType/{eventTypeID}` – Hent hendelser etter type

Quiz

- `POST /api/quizzes` – Start ny quiz
- `POST /api/quizzes/{quizId}/answers` – Send inn svar
- `GET /api/quizzes/{quizId}` – Hent detaljer for quiz-forsøk
- `GET /api/quizzes/{quizId}/result` – Hent quiz-resultat
- `GET /api/quizzes/history/{userId}` – Hent quiz-historikk for bruker

Refleksjoner

- `POST /api/reflection/add` – Legg til refleksjonsnotat
- `GET /api/reflection/public` – Hent alle offentlige refleksjonsnotater
- `GET /api/reflection/my` – Hent mine refleksjonsnotater
- `GET /api/reflection/household` – Hent husstandens refleksjonsnotater
- `DELETE /api/reflection/{id}` – Slett refleksjonsnotat

Websocket

- STOMP-endepunkt: `/ws` - websocket-endepunkt
- `ws/topic/events` - Nye eller oppdaterte hendelser
- `ws/topic/events/delete` - Slettede hendelser
- `ws/topic/positions` - Nye eller oppdaterte posisjoner
- `ws/topic/positions/delete` - Slettede posisjoner
- `ws/topic/notifications` - Nye varsler
- `ws/topic/notifications/delete` - Slettede varsler

Registrering

Når nye brukere registrerer seg, må de verifisere e-postadressen sin. Samt fullføre en recaptcha-verifisering for å hindre

spam og automatiserte registreringer.

Passord

Passordene lagres i databasen som en hash. Det er ikke mulig å hente ut passordet fra databasen. Passordene må

være minst 8 tegn lange og inneholde både store og små bokstaver, samt tall og spesialtegn. Til hashing av passordene

brukes bcrypt.

Authentisering

Når brukeren logger inn eller registrerer seg, får de en JWT-token som lagres i HTTP-only cookie. Tokenen sendes med

hver forespørsel til serveren for å autentisere brukeren.

Autorisering

Autorisering håndteres av spring security sin SecurityFilterChain. Der den tillater spesifikke adresser som localhost:5173

og gitlab. Videre defineres hvilke endepunkter som kan nås uten autentisering og hvilke som krever autentisering.

Avhengighetssjekk

For å sikre at applikasjonen ikke har kjente sårbarheter, brukes OWASP Dependency-Check. Dette verktøyet skanner

avhengighetene i prosjektet og rapporterer eventuelle kjente sårbarheter. Det finnes mer informasjon om når denne

skanningen kjøres under [Home/System/CI-og-Testing]

Injeksjon

Ved å bruke JPA og Hibernate for databaseoperasjoner, reduseres risikoen for SQL-injeksjoner. JPA håndterer parameterbinding og sanitiserer input automatisk. Vue og Spring Boot er også beskyttet mot XSS-angrep ved å bruke innebygde mekanismer for å håndtere brukerinput og unngå skriptingangrep. mer informasjon finnes her: [\[https://docs.spring.io/spring-security/reference/servlet/exploits/headers.html#servlet-headers-xss-protection\]](https://docs.spring.io/spring-security/reference/servlet/exploits/headers.html#servlet-headers-xss-protection)

5.1 Brukertester

Formål

Det overordnede målet for den første brukertesten er å finne ut om systemet er lett navigerbart, og om brukere forstår

hvordan de skal bruke funksjonalitet systemet tilbyr utifra designet. Finne ut hvor brukervennlig nettsiden er.

Deltakere

- **Linus:** har god erfaring med digitale tjenester og kan regnes som teknologisk kompetent.

Oppsett

- Testen utføres på en laptop i Google Chrome.
 - **Testobservatør:** Jonathan Hübertz tar notater.
 - **Testdeltaker:** Linus møter opp fysisk for å ta testen.
-

Oppgaver og observasjoner

| ID | Oppgave | Suksesskriterier | Observasjoner |
|----|--|----------------------|--|
| | - Fullfører registrering | | |
| | Bruker greide å registrere ny | | |
| | Registrer ny bruker. Du er på under 2 min. | | |
| 1 | en ny bruker på plattformen - Forstår eventuelle | | bruker. Prøvde å åpne via e-postklient, men dette fungerer |
| | og vil registrere en ny | feilmeldinger og hva | |
| | ikke; Outlook fungerer heller | | |
| | bruker. | som må endres for å | |
| | ikke. | | |
| | registrere bruker. | | |
| | Opprette husstand etter e- | | |
| | - Klarer å opprette en ny | | |
| | postverifisering av bruker. | | |
| 2 | | | husstand på under 2 Dette gikk helt fint. |
| | Forstår layoutet av «opprett | | |
| | min. | | |
| | husstand»-siden og feltene. | | |
| | - Bruker finner frem til | | |
| | Fant beredskapslager enkelt | | |
| | beredskapslager på 1 | | |
| | Finne frem til | | etter å ha lest infotekst; teksten |

3

beredskapslager.

hovedmeny med «før,

lese.

under, etter».

Redigere beredskapslager.

- Bruker klarer å legge til

Bruker får beskjed om å

4

legge til 10 l vann og 2

min.

jodtabletter.

- Finn kartet på under 30 Litt lite intuitivt; vanskelig å se

Finne og bruke kartet for å sek.

5

finne nærmeste tilfluktsrom. - Identifiser korrekt

nærmeste tilfluktsrom. lageret.

- Finn

husholdningssiden på

Generere invitasjonskode til under 30 sek.

6

sin husholdning.

- Forstå hva som er

invitasjonskoden i løpet

av 30 sek.

ID

Oppgave

Suksesskriterier

Observasjoner

- Logge ut på under 30

Logge ut, og så logge inn sek fra hovedmeny.

7

min. Start fra

burde vært større og lettere å

riktige varer på under 1

Gikk på første forsøk.

markørene når man er zoomet

inn. Krunglete navigasjon
fra

Gikk helt fint.

30 Gikk helt fint.

| | | |
|------------------------------|------------------------|--|
| igjen. | - Logge inn på under | |
| sek etter utlogging. | | |
| - Finn | | |
| husholdningssiden på | | |
| Opprette et nytt | under 30 sek. | |
| 8 | | refleksjonsnotat med - Opprett et notat med Gikk helt fint. |
| offentlig synlighet. | vilkårlig innhold og | |
| offentlig synlighet på 30 | | |
| sek. | | |
| Finne offentlig | | Gikk fint; brukte litt tid på å |
| - Bruk søkefeltet og finn | | |
| 9 | | refleksjonsnotat med ordet oppdage søkefeltet, men rakk |
| notatet på under 30 sek. | | |
| «happiness» i innholdet. | | det innen tiden. |
| - Finn «Dine notater» → | | |
| 10 | | Slette refleksjonsnotatet sitt. Gikk helt fint. klikk notatet → slett. |
| - Naviger til quiz-siden | | |
| Gjennomføre en | | |
| på under 1 min. | | |
| 11 | | beredskapsquiz på 7 Det gikk fint. |
| - Endre antall spørsmål | | |
| spørsmål. | | |
| fra 5 til 7. | | |
| - Trykk logo/krisefikser- | | |
| Gå tilbake til hjemmeside og | header for å komme til | |

12

finne tilbake til svarene fra
hjemmsiden. Gikk veldig
fint.

quizen i punkt 11. - Finn quiz-svarene på
hvert spørsmål.

- Finn bjelle-ikonet i

13

Se sine notifikasjoner.
navbaren på under 30
Gikk veldig fint.

sek.

Se hendelser med moderat *Suksesskriterier ikke*

14

Gikk veldig fint.

status på kartet. *spesifisert.*

Formål

Det overordnede målet for den første brukertesten er å finne ut om systemet er lett navigerbart, og om brukere forstår

hvordan de skal bruke funksjonalitet systemet tilbyr utifra designet. Finne ut hvor brukervennlig nettsiden er.

Deltakere

- **Fredrik:** har god erfaring med digitale tjenester og kan regnes som teknologisk kompetent.

Oppsett

- Testen utføres på en laptop i Google Chrome.
- Scott du Plessis er testobservatør og tar notater.
- Fredrik møter opp fysisk for å ta testen.

Oppgaver og observasjoner

| ID | Oppgave | Suksesskriterier | Observasjoner |
|----|--|---|--|
| | | - Fullfører registrering på under 2 min. Burde vise at passord fortsatt er feil - Forstår eventuelle når man trykker registrer etter å ha feilmeldinger og hva prøvd å endre passordet til noe som må endres for å kunne gyldig. registrere bruker. | |
| 1 | Registrer ny bruker. Du er en ny bruker på plattformen og vil registrere en ny bruker. Starter på hovedmeny. | - Klarer å opprette en ny husstand på under 2 min. | Alt gikk fint. |
| | Opprette husstand etter epost verifisering av bruker. 2 Forstår layoutet av «opprett husstand»-siden og hvilke felter som må fylles inn. | - Bruker finner frem til beredskapslager på 1 min. Start fra hovedmeny med «før under etter». | Ingenting tilsier hvor beredskapslager ligger (før, under, etter). Ingen av boksene forklarer hva man finner inni de. |
| 3 | Finne frem til beredskapslager. | - Bruker klarer å legge til riktig varer på under 1 min. | Hele boksen for gjenstanden (vann for eksempel) burde være clickable for å redigere mengde. Fredrik trodde at «rediger»-knappen var for å endre navnet på varen. Fant lett frem til kartet ved å prøve |
| 4 | Redigere beredskapslager. Bruker får beskjed om å legge til 10 liter vann og 2 jodtabletter. | - Bruker finner kart på under 30 sek. - Identifiserer korrekt nærmeste tilfluktsrom. - Finnhusholdningssiden på under 30 sek. | seg frem. Vanskelig å se tilfluktsrommene pga. kontrast (WCAG); ikonet kan gjøres større. |
| 5 | Finne og bruke kartet for å finne nærmeste tilfluktsrom. | - Forstår hva som er invitasjonskoden i løpet av 30 sek. - Logge ut på under 30 sek fra hovedmeny. | Gikk helt fint, ingen problemer. |
| 6 | Generere invitasjonskode til sin husholdning. | - Logge inn på under 30 sek etter utlogging. | Gikk helt fint, ingen problemer. |
| 7 | Logge ut, og så logge inn igjen. | | |

title: Sprint 1

Plan sprint 1

Periode: 22.04.2025 - 29.04.2025

Hva var målet for sprinten?

Sprinten hadde som mål å legge fundamentet for applikasjonen

Sprint 1-Review

Dato: 29.04.2025

Vi begynte med å vise en demo av produktet til produkteier (PO). De syntes at produktet generelt sett så bra ut basert på

hvor vi lå i prosessen. De syntes vi har kommet godt i gang.

Etter demoen spurte vi dem hva de ønsket at vi skulle prioritere fremover. De svarte at de ønsket at vi skulle få til en quiz

og kalkulering av beredskapsscore. De sa også at nettsiden burde tilby informasjon som "motiverer" ikke-registrerte

brukere til å registrere seg.

Vi spurte også hvordan de ønsket at man skulle oppdatere beredskapslageret. Her sa de at de skulle komme tilbake til

oss med et godt svar på dette.

PO-ene nevnte også at dette ønsket en ordentlig personvernserklæring, samt muligheten for en bruker å kunne forlate

en husstand.

Vi etterspurte også hvilket fargepalett de ønsket på nettsiden. Her ba de oss være kreative

Plan Sprint 2

Periode: 29.04.2025 - 9.05.2025

Hva var målet for sprinten?

Sprinten hadde som mål å fullføre all kildekoden til prosjektet.

Sprint 1-Review

Dato: 29.04.2025

Vi begynte med å gå gjennom user stories og akseptanserkriteriene og P.O. vurderte om kriteriene var oppfylt.

Under gjennomgangen påpekte P.O. at vi burde skrive mer på "etter krise" på det generelle innholdet.

P.O. synes også at for få tilfluktsom viste så vi bestemte at vi skal legge inn tilfluktsrom fra geogjson fil.

P.O. ønsket at grad av beredskap skal vises i uker, dager og timer istedenfor en prosent.


P.O. påpektet at vi ikke har full vibrasjon og lyd løsning på ting men synes det går helt bra ettersom det er en

webapplikasjon og ikke en app.

Avslutningsvis uttrykte produkteier stor tilfredshet med innsatsen og fremdriften, og ingen flere tilpasninger er

nødvendige.

6 WCAG-matrise

| | |
|--|--|
| Sist oppdatert: 24.01.2024 | Oversikt over status for universell utforming i én enkelt nettløsning. Her ser du WCAG-kravene som nettstedet skal oppfylle, sortert på de fire prinsippene i WCAG-standarden. For hver av retningslinjene finnes beskrivelser, lenker til mer informasjon, m.m., samt en nedtrekksmeny der du kan notere status pr i dag. |
|  uutilsynet | Du kan laste ned arkiet og bruke det som arbeidsliste, men følg alltid med på uutilsynet.no for oppdatert kravliste. |

Mal – WCAG-sjekkliste for utfylling av tilgjengelighetserklæring for nettløsning

| | Retningslinje | Beskrivelse av retningslinje | Suksesskriterium | Følges kravet? | Svaret som skal avgis |
|---------------------------------|--|---|---|---------------------------------|--|
| Prinsipp 1: Mulig å oppfatte | 1.1 Tekstalternativer | Gi tekstalternativer til alt ikke-tekstlig innhold, slik at det kan konverteres til formater som brukere har behov for, for eksempel stor skrift, | 1.1.1 Ikke-tekstlig innhold | Ja | |
| | 1.2 Tidsbaserte medier (lyd og video) | Gi alternativer til tidsbaserte medier. | 1.2.1 Bare lyd og video | Vi har ikke denne typen innhold | |
| | | | 1.2.2 Teksting (forhåndsinnspilt) | Vi har ikke denne typen innhold | |
| | | | 1.2.5 Synstolkning (forhåndsinnspilt) | Vi har ikke denne typen innhold | |
| | 1.3 Mulig å tilpasse | Lag innhold som kan presenteres på forskjellige måter uten at informasjon eller | 1.3.1 Informasjon og relasjoner | Nei | Burde bruke h1-h6 elementer der det passer |
| | | | 1.3.2 Meningsfylt rekkefølge | Ja | |
| | | | 1.3.3 Sensoriske egenskaper | Nei | Står kun "Feltet må fylles ut" når man ikke fyller et felt, burde fortelle hvilket felt(loginn register) |
| | | | 1.3.4 | Nei | Usikker, burde dobbeltsjekkes |
| | | | 1.3.5 Identifiser formål med inndata | Nei | Sikjente ikke kravet |
| | 1.4 Mulig å skille fra hverandre | Gjør det enklere for brukere å se og høre innhold, blant annet ved å skille forgrunnen fra | 1.4.1 Bruk av farge | Ja | |
| | | | 1.4.2 Styring av lyd | typen innhold | |
| | | | 1.4.3 Kontrast (minimum) | Ja | |
| | | | 1.4.4 Endring av tekststørrelse | | |
| | | | 1.4.5 Bilder av tekst | Ja | |
| | | | 1.4.10 Dynamisk tilpasning (Reflow) | Ja | |
| | | | 1.4.11 Kontrast for ikke-tekstlig innhold | Nei | Logo skiller ikke godt nok fra navbar farge. Kan fikses ved å ha en hvit linje rundt logo |
| | | | 1.4.12 Tekstavstand | Ja | |
| | | | 1.4.13 Pekerfølsomt innhold eller innhold ved | Ja | |
| Prinsipp 2: Mulig å betjene | 2.1 Tilgjengelig med tastatur | Gjør all funksjonaliteten tilgjengelig med tastatur. | 2.1.1 Tastatur | Ja | |
| | | | 2.1.2 Ingen | Ja | |
| | | | 2.1.4 Hurtigtaster som består av ett | Vi har ikke denne typen innhold | Vi har ingen hurtigtaster som består av ett tegn |
| | 2.2 Nok tid | Gi brukere nok tid til å lese og bruke innhold. | 2.2.1 Justerbar hastighet | Ja | |
| | | | 2.2.2 Pause, stopp | Ja | |
| | 2.3 Anfall og andre fysiske reaksjoner | Ikke utform innhold på en måte som er kjent for å forårsake | 2.3.1 Terskelverdi på maksimalt tre glimt | Vi har ikke denne typen innhold | Ikke noe blinkende innhold |
| | 2.4 Navigerbar | Gjør det mulig for brukere å navigere, finne innhold og vite hvor de befinner seg. | 2.4.1 Hoppe over blokker | Nei | |
| | | | 2.4.2 Sidetitler | Ja | |
| | | | 2.4.3 | Ja | |
| | | | 2.4.4 Formål med lenke (i kontekst) | Ja | |
| | | | 2.4.5 Flere måter | Nei | |

| | | | | | |
|------------------------|--------------------|---|---|---------------------------------|---|
| | | | 2.4.6 Overskrifter og ledetekster | Ja | |
| | | | 2.4.7 Synlig fokus | Nei | "Før under etter" bokser har ikke det. Samme med knapper på husstandsside |
| | 2.5 Inndatametoder | Gjør det enklere for brukerne å betjene funksjonalitet med andre inputmetoder enn bare | 2.5.1 | Ja | |
| | | | 2.5.2 | Ja | |
| | | | 2.5.3 Ledetekst i | Nei | Skjente ikke kravet |
| Prinsipp 3: Forståelig | | | 2.5.4. Bevegelsesaktivering | Vi har ikke denne typen innhold | |
| | 3.1 Leselig | Det må være mulig å forstå informasjon og betjening av brukergrensesnitt. | | | |
| | | | 3.1.1 Språk på siden | Ja | |
| | | | 3.1.2 Språk på deler av innhold | Ja | |
| | 3.2 Forutsigbar | Sørg for at websider presenteres og fungerer på | | | |
| | | | 3.2.1 Fokus | Ja | |
| | | | 3.2.2 Inndata | Nei | Skjente ikke kravet |
| | | | 3.2.3 Konsekvent navigering | Ja | |
| | | | 3.2.4 Konsekvent identifikasjon | Ja | |
| | 3.3 Inndatahjelp | Hjelp brukere med å unngå feil og å rette opp feil. | | | |
| | | | 3.3.1 Identifikasjon av feil | Ja | |
| | | | 3.3.2 Ledetekster eller instruksjoner | Ja | |
| | | | 3.3.3 Forslag ved | Ja | |
| | | | 3.3.4 Forhindring av feil (juridiske feil, økonomiske feil, datafeil) | Nei | Kan ikke angre innsending av registrering |
| Prinsipp 4: Robust | 4.1 Kompatibel | Sørg for best mulig kompatibilitet med aktuelle og fremtidige brukere, inkludert kompensering | | | |
| | | | 4.1.1 Parsing | Ja | |
| | | | 4.1.2 Navn, rolle,... | Nei | Tror ikke det, burde dobbeltsjekkes |
| | | | 4.1.3 | Ja | |

7 Sprint planlegging referater

Møtereferat: 2. Sprint Planning

Dato: **29.04.25**

Deltakere: **Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray**

Produkteiere deltatt i møtet: **Kristian Storehaug, Zakariya Ibrahim**

Møteleder: **Scott du Plessis**

Møtereferent: **Aryan Malekian**

Agenda:

Panelgge kommende (2) sprint.

Formål:

Bestemme hvilke mål som skal prioriteres for 2. sprint i samråd med produkteiere.

Diskusjon og prioriteringer

P.O. nevner at følgende måloppgaver må prioriteres i 2. sprint:

- Quiz skal prioriteres
- Kalkulering av beredsskapsbeholdning må på plass
- Brukeren må kunne redigere egne beredskapsitems på en brukervennlig måte
- Personvernserklæring skal implementeres
- Generell informasjon skal være synlig for ikke-innloggede brukere
- Det skal utvikles motiverende innhold på forsiden som fremmer registrering

Avklaringer fra teamet og svar fra P.O.:

Skal ikke-registrerte brukere ha tilgang til innhold?

- P.O.: Nei, brukeren skal redirectes til login umiddelbart.

Skal bruker kunne forlate eller bytte husstand?

- P.O.: Muligens, men lav prioritet. Kanskje husstandsleder kan fjerne brukere.

Hvordan skal informasjon struktureres «før, under og etter krise»?

- P.O. ga svar: Generell informasjon skal vises på hver av seksjonene, ellers står det fritt til at teamet bestemmer dette selv.

Tekniske føringer P.O.:

- Teamet må koble frontend og backend løpende, og unngå å utsette integrasjon-
- Forespørsel om kreativ frihet til fargevalg ble gjentatt.

8 sprint retrospektiv 1 referat

Møtereferat: 1. Sprint Retrospective

Dato: **29.04.25**

Deltakere: **Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray**

Møteleder: **Scott du Plessis**

Møtereferent: **Aryan Malekian**

Agenda:

Teammedlemmene skal oppsummere hvordan de har opplevd en foregående sprinten og dele deres følelsesregister.

Formål:

Målet er at teammedlemmene skal forstå sin prestasjon bedre, og danne et grunnlag for å kunne forbedre seg til neste sprint.

Startøvelse

Scrum Master ber hvert teammedlem oppsummere deres opplevelse av forrige sprint med ett ord.

Jonathan: Fundament

Usman: Uoversiktlig

Mikael: Kaos

Sander Berge: Lærerikt

Sander Nessa: Lærerikt

Aryan: Gøy

Hva likte vi?

Mikael: «Vi kom i gang». Mikael er positiv til at teamet har klart å levere en MVP.

Scott: Enig med Mikael, likte godt at teamet fikk satt i gang arbeidet for alvor og begynt å se resultater.

Usman: Enig med de andre så langt, ser at det har vært progresjon og er fornøyd med det.

Jonathan: Fornøyd med teamet har kommet inn i rutine.

Hva har vi lært?

Scott: «Jo flere kokker, jo mer søl». Til forskjell fra det forrige prosjektet, i emnet Full-Stack Applikasjonsutvikling, har vi nå enda flere medlemmer i teamet. Dette

har ført til at arbeidet har blitt litt mer kaotisk, og til tider at det har vært vanskelig å forholde seg til arbeidet som har blitt gjort av (de mange) andre på teamet.

Sander Berge: Har lært at det er viktig å forså visjonsdokumentet bedre så tidlig som mulig i prosjektet.

Scott: Har fått bedre innsikt i hvordan utvikling skjer i næringslivet.

Aryan: Sier seg enig med Scott og Sander Nessa.

Hva kunne vært bedre?

Usman: Kunne hatt bedre oversikt over prosjektet – hva som er gjort av andre teammedlemmer i løpet av sprinten, samt å ha bedre oversikt over prosjektdokumentene for øvrig.

Scott & Jonathan: Vi må dele opp issues i mindre deler! Hver issue på forrige Sprint Backlog var for stor, og det ble til tider vanskelig å føle på progresjon.

Sander Nessa: Vi manglet *Burndown Chart* – dette lages nå.

Mikael: Vi burde ha bestemt oss for en mappestruktur og teknisk arkitektur fra starten av i prosjektet.

Usman: Uklarhet om innhold på ulike sider. Vi bør avklare (innad i teamet gjennom PO-Proxy) hva hver seksjon (før, under, etter) skal inneholde av informasjon.

Aryan: Vi må bli flinkere til å dokumentere spørsmål til P.O. løpende, slik at vi vet hva vi skal spørre om under møter.

Kommentarer fra teamet:

- P.O. burde vært bedre orientert om innholdet i visjonsdokumentet.
- Tilbakemeldinger og instruksjoner fra P.O. har vært **vage**, noe som har skapt usikkerhet underveis i utviklingen.

9 sprint retrospektiv 2 referat

Møtereferat: 2. Sprint Retrospective

Dato: **09.05.25**

Deltakere: **Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge**

Møteleder: **Scott du Plessis**

Møtereferent: **Aryan Malekian**

Agenda:

Reflektere over sprinten som nettopp ble gjort. Identifisere hva som var bra og hva som kunne gått bedre.

Formål:

Tillate alle å dele sitt følelsesregister, samt få muligheten til å finne forbedringsmomenter individuelt og som et team.

Spørsmål: Hvis vi skulle gjort sprinten igjen, hva ville vi gjort annerledes?

Usman:

Er fornøyd med hvordan sprinten gikk og ville ikke endret på noe.

Scott:

Er uenig – mener det var forbedringspotensial på flere områder.

Teamets refleksjoner:

- Arbeidsoppgaver kunne vært fordelt jevnere mellom teammedlemmene.
- Vi burde pullet og merget oftere – flere små pull requests i stedet for store.
- Flere issues burde vært delt opp i mindre deler for bedre flyt.
- Det burde vært planlagt alternativt arbeid for de som ble ferdige tidlig.
- Frontend burde vært testet løpende, ikke først mot slutten.
- Issues kunne vært tydeligere merket med navn og ansvar.
- Vi skulle ha satt oss bedre inn i visjonsdokumentet tidlig.
- Databaseløsningen kunne vært mer robust – f.eks. ved å bruke Flyway.

Spørsmål: Hva er vi fornøyd med?

- God testdekning og systematisk arbeid med testing på backend.
- Unngikk skipptak – arbeidet har vært jevnt fordelt over tid.
- Jevn arbeidsinnsats og god balanse mellom fokus og pauser.
- Flat teamstruktur der alle har blitt hørt og tatt på alvor.
- God stemning, godt samhold – teamet har hatt det gøy sammen.
- Flere uttrykte at de har gledet seg til å komme på skolen.
- Vi har vært lydhøre for P.O.-s ønsker og vist kreativitet i fravær av spesifikke føringer.

Teamet uttrykker at de stort sett er fornøyd med egen innsats og sluttresultatet.
Alle er slitne og klare til å bli ferdig.

10 sprint review 1 referat

Møtereferat: Første Sprint Review

Dato: **29.04.25**

Deltakere: **Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray**

Produkteiere deltatt i møtet: **Kristian Storehaug, Zakariya Ibrahim**

Møteleder: **Scott du Plessis**

Møtereferent: **Aryan Malekian**

Agenda:

Teamet presenterer en demo av det som er utviklet så langt.

Formål:

Gjennomgang av fremdrift og hva som ble levert i første sprint.

Tilbakemeldinger fra P.O.:

Alt ser bra ut så langt – teamet har kommet langt. Brukere må logge inn for å få tilgang – ikke-registrerte brukere skal ikke ha full tilgang.

P.O. oppfordrer teamet til å være kreative og velge en innbydende fargepalett til applikasjonen, ettersom den gjeldende paletten er kjedelig.

Videre ønsker P.O. en avklaring på om en person i husholdning kan invitere andre. Det nevnes noe om å ha en «husfar» (superadmin), men vi bestemmer oss for å ta opp dette i sprint planning.

P.O. ønsker mulighet for superadmin (her tolker P.O. superadmin som en «husfar» som kan ha spesielle rettigheter i en husholdning, ikke superadmin slik det er beskrevet i visjonsdokumentet).

11 Sprint review 2 referat

Møtereferat: 2. Sprint Review

Dato: **09.05.25**

Deltakere: **Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge**

Produkteiere deltatt i møtet: **Kristian Storehaug**

Møteleder: **Scott du Plessis**

Møtereferent: **Aryan Malekian**

Agenda:

Teamet presenterer en demo av det som er utviklet så langt.

Formål:

Gjennomgang av det teamet har levert i sprinten, vise en demo til P.O. Kristian og motta tilbakemeldinger.

Demonstrert funksjonalitet og status:

- Systemet må støtte visning av tilfluktsrom hentet fra en GeoJSON-fil.
- Teamet har klart å visualisere tilfluktsrom på kartet, noe både teamet og P.O. oppfatter som vellykket.
- Arbeid på "Etter krise"-delen pågår, og det ble nevnt at mer generelt innhold bør legges inn.
- Graden av beredskap vises nå som uker, dager og timer, noe P.O. oppfatter som nyttig.
- Det er ikke full støtte for vibrasjon og lydvarsling, men P.O. mener det er helt greit og ikke et krav for nåværende leveranse.

Tilbakemelding fra Product Owner:

- P.O. uttrykker at teamet har vært flinke og oppegående gjennom hele sprinten, men også over hele prosjektet.
- Generell oppfatning fra P.O. er at ingenting trenger å endres på nåværende tidspunkt, og vi er klare til å levere.
- P.O. er fornøyd med fremdriften og resultatet slik den står.

12 Møtereferat: 1. Scrum Standup

Dato: 23.04.25

Klokkeslett: 09:00

Deltakere: Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray

Møteleder: Scott du Plessis

Møtereferent: Aryan Malekian

Agenda:

- Hver deltaker informerer om hva de gjorde forrige dag, hva de planlegger å gjøre i dag, og om de har noen blokkeringer.
- Felles status for dagens mål og prioriteringer

Formål:

Få oversikt over fremdrift i prosjektet, fordele dagens arbeidsoppgaver og identifisere eventuelle hindringer i teamet.

Deltakernes status:

Jonathan Hubertz

I går: Satt opp Spring og Vue. Jobbet med wireframe sammen med Aryan. Bidro i utarbeidelsen av backlog.

I dag: Skal jobbe videre med wireframe og starte på opprettelse av issues i GitLab.

Sander Nessa:

I går: Jobbet med utarbeidelse av backlog.

I dag: Skal sette opp issues og begynne å jobbe med implementering av første brukerhsitorier.

Sander Berge

I går: Jobbet med backlog.

I dag: Planlegger å starte teknisk arbeid med funksjonalitet i henhold til prioriterte oppgaver.

Usman Ghafoorzai

I går: Opprettet GitLab.-repositorium. Bidro til backlogarbeid.

I dag: Skal starte arbeid med issue sog begynne utvikling i backend.

Mikael Stray

I går: Fikk ikke jobbet aktivt, men leste gjennom prosjektoppgaven under reise.

I dag: Vil orientere seg mer om prosjektets tekniske og funksjonelle krav for å komme i gang.

Aryan Malekian

I går: Jobbet med wireframes sammen med Jonathan. Bidro også til utarbeidelsen av produkt backlog.

I dag: Jobbe videre med wireframes, eventuelt påbegynne arbeid med kartfunksjonalitet.

Felles diskusjon:

- **Behov for wireframe:**

Teamet er enige om at det må utarbeides en felles wireframe som grunnlag for frontendarbeidet. Arbeidet er påbegynt.

- **Avklaring rundt prioriteringer:**

Det uttrykkes behov for tydeligere instruksjoner om hvordan backlog items skal prioriteres. Teamet avventer svar eller retning fra P.O.

Plan for dagen:

- **Sette opp Spring-backlog i GitLab**
- **Lage og fordele issues**
- **Starte med login – og registreringsfunksjonalitet**
- **Etablere test- og utviklingsmiljø**
- **Implementere JWT-token-autentisering**

6. Daily scrum 2 referat

13 Møtetreferat: 2. Scrum Standup

Dato: 24.04.25

Klokkeslett: 09:00

Deltakere: Scott du Plessis, Aryan Malekian, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray

Møteleder: Scott du Plessis

Møterefereent: Aryan Malekian

Agenda:

- Hver deltaker informerer om hva de gjorde forrige dag, hva de planlegger å gjøre i dag, og om de har noen blokkeringer.
- Felles status for dagens mål og prioriteringer

Formål:

Opprettholde oversikt og fremdrift i sprinten, sikre koordinasjon i teamet og justere kurs ved behov.

Deltakernes status:

Usman Ghafoorzai

I går: Ryddet i Vue-strukturen, fjernet unødvendige filer. Lagde issue sog organiserte issue board. Startet utvikling av HomepageView, inkludert containerstruktur og reusable komponenter.

I dag: Skal begynne på implementering av navbar.

Sander Nessa

I går: Deltok i arbeidet med å opprette issues. Jobbed med grunnstruktur i backend, blant annet ved å lage endpoints for item-kategorier.

I dag: Skal sette opp GitHub-repo og se på løsninger for logging og testing.

Mikael Stray

I går: Arbeidet med issue sog implementerte sider for login, registrering og glemt passord.

I dag: Skal starte utvikling av admin-funksjonalitet.

Scott du Plessis:

I går: Opprettet issues og gjenbrukte relevante komponenter fra tidligere fullstack-prosjekt. Satte opp logging, backend-funksjonalitet og startet på registrer-bruker-funksjon.

I dag: Fortsetter med utvikling relatert til registrering og backend-integrasjon.

Sander Berge

I går: Arbeidet med å sette opp prosjektmiljøet lokalt. Startet på utvikling av wireframes.

I dag: Skal fortsette arbeidet med wireframes.

Aryan Malekian

I går: Jobbet med kartfunksjonalitet og startet planlegging av lagerstruktur sammen med Jonathan.

I dag: Fortsetter utviklingen av kartet og arbeidet med lagerløsning.

Jonathan Hubertz

Status: Ikke til stede (meldt fravær grunnet fysioterapi).

Felles diskusjon:

- **Behov for wireframe:**

Teamet er enige om at det må utarbeides en felles wireframe som grunnlag for frontendarbeidet. Arbeidet er påbegynt.

- **Avklaring rundt prioriteringer:**

Det uttrykkes behov for tydeligere instruksjoner om hvordan backlog items skal prioriteres. Teamet avventer svar eller retning fra P.O.

Plan for dagen:

- **Sette opp Spring-backlog i GitLab**
- **Lage og fordele issues**
- **Starte med login – og registreringsfunksjonalitet**

- Etablere test- og utviklingsmiljø
- Implementere JWT-token-autentisering

7. daily scrum 3 referat

14 Møtereferat: 3. Scrum Standup

Dato: 25.04.25

Klokkeslett: 09:00

Deltakere: Scott du Plessis, Aryan Malekian, Jonathan Hubertz, Sander Nessa, Usman Ghafoorzai, Sander Berge, Mikael Stray

Møteleder: Scott du Plessis

Møtereferent: Aryan Malekian

Agenda:

- Hver deltaker informerer om hva de gjorde forrige dag, hva de planlegger å gjøre i dag, og om de har noen blokkeringer.
- Justere innsats og sikre fremdrift.

Formål:

Opprettholde oversikt og fremdrift i sprinten, sikre koordinasjon i teamet og justere kurs ved behov.

Deltakernes status:

Sander Nessa

I går: Satt opp entity for items, laget endpoints for å hente ut alle kategorier, en spesifikk kategori og alle items. Startet også med å lage tester.

I dag: Fortsetter med testing og videreutvikling av item- og kategorilogikk.

Usman Ghafoorzai

I går: Gjorde justeringer på HomepageView for å tilpasse den bedre til wireframes. Fullførte navbar, og begynte på utvikling av «Under Krise»-siden.

I dag: Fortsetter med utviklingen av «Under Krise»-visningen og eventuell kobling mot data.

Sander Berge

I går: Arbeidet videre med wireframes, som nå er på et tilfredsstillende nivå. Startet på «Etter krise»-siden, som nærmer seg ferdig.

I dag: Ønsker å starte med utvikling av admin-siden.

Jonathan Hubertz

I går: Satte opp DTOs for storage, jobbed med API-klienter, og designet strukturen for lageret.

I dag: Skal fjerne Vuetify fra prosjektet og implementere funksjonalitet for å oppdatere lageret.

Scott du Plessis

I går: Fullførte funksjonalitet for login og registrering på backend. Startet utvikling av «Join Household»-funksjon, men rakk ikke å gjøre den ferdig. La også til tester.

I dag: Har som mål å ferdigstille «Join Household»-funksjonaliteten og forbedre testdekningen.

Aryan Malekian

I går: Jobbet med videre med kartfunksjonalitet. Byttet fra Google Maps til Leaflet-map, for å unngå eventuelle problemer med API-nøkkel og andre GMap konfigurasjoner (som lokal Google DOM) under utvikling.

I dag: Fortsetter utviklingen av kartet, med fokus på å vise shelters.

8. daily scrum 4 referat

15 Møtereferat: 4. Scrum Standup

16 Møtereferat: 5. Scrum Standup

17 Møtereferat: 6. Scrum Standup

18 Møtereferat: 7. Scrum Standup

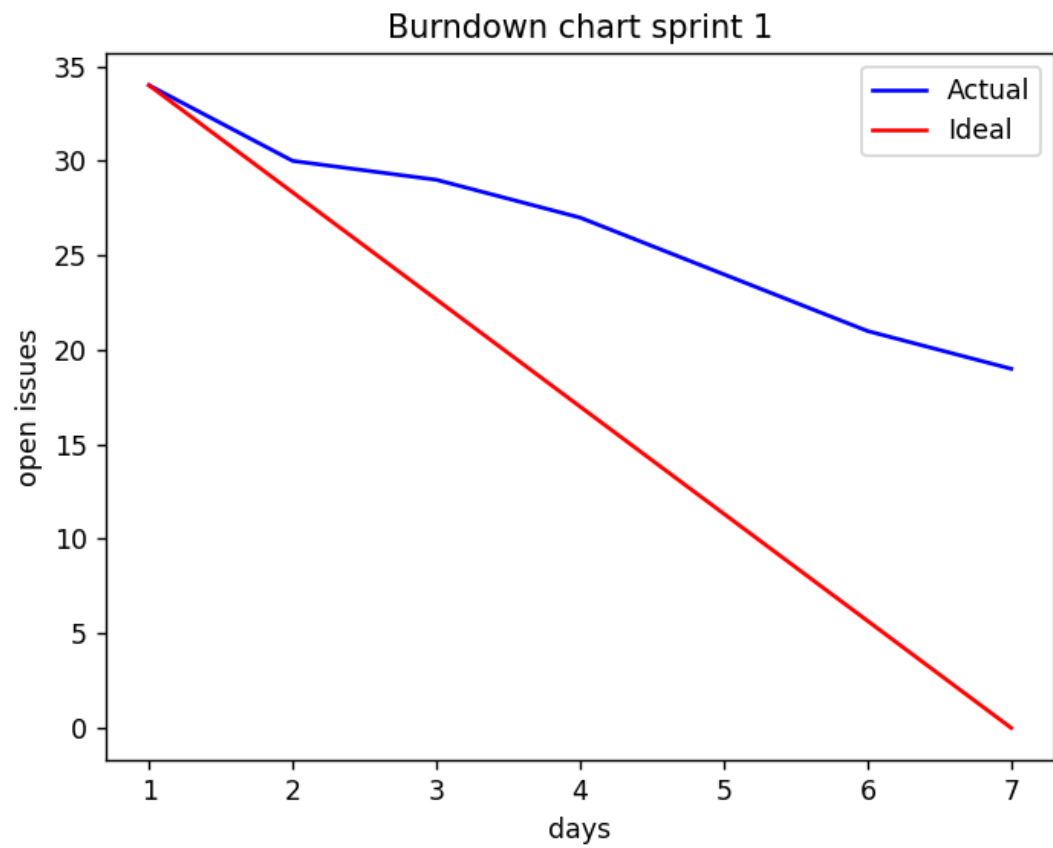
19 Møtereferat: 8. Scrum Standup

20 Møtereferat: 9. Scrum Standup

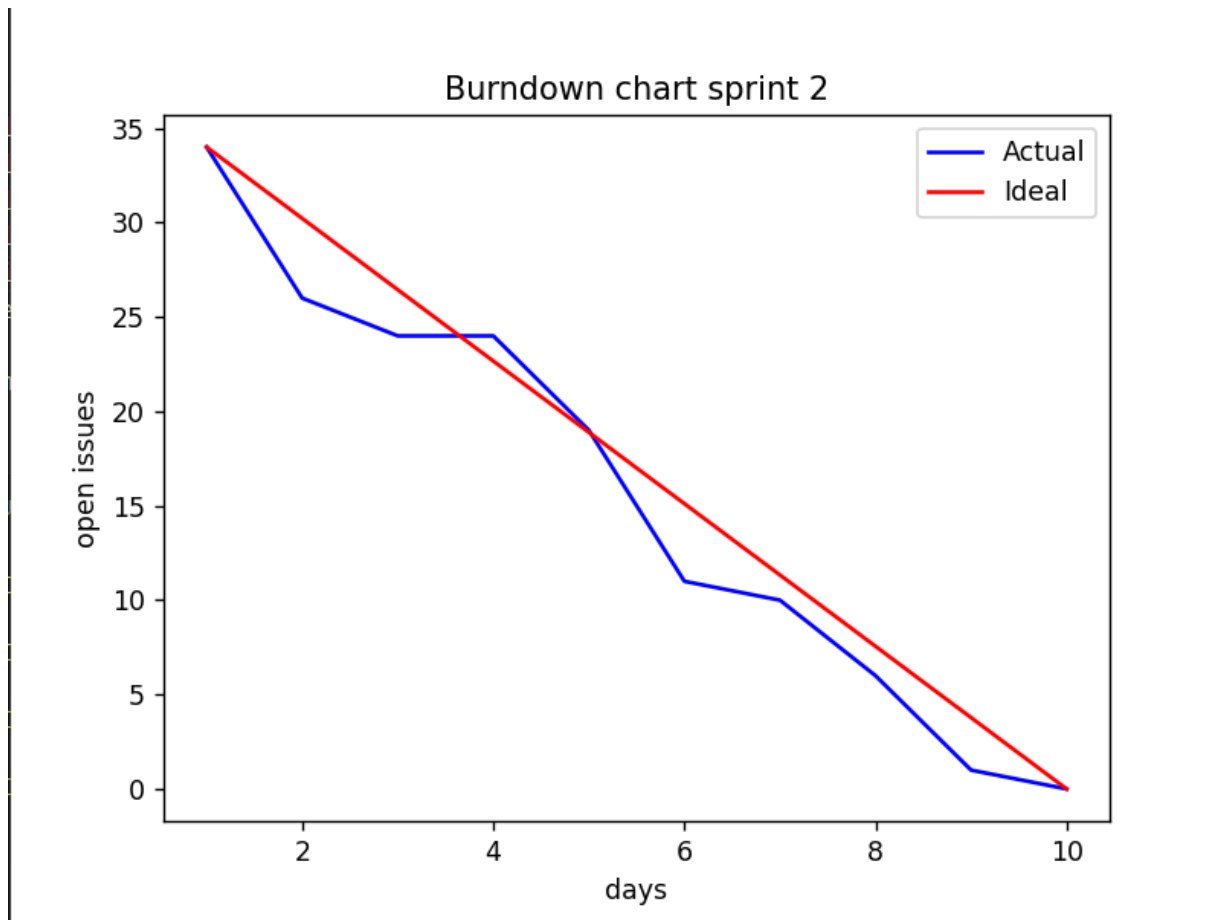
21 Møtereferat: 10. Scrum Standup

22 Møtereferat: 11. Scrum Standup

23 Burndown chart sprint 1



24 burndown chart sprint 2



25 OWASP Dependency check



Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or other or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

[How to read the report](#) | [Suppressing false positives](#) | [Getting Help: github issues](#)

♡ [Sponsor](#)

Project: frontend

Scan Information ([show all](#)):

- *dependency-check version:* 12.1.1
- *Report Generated On:* Fri, 9 May 2025 09:41:40 GMT
- *Dependencies Scanned:* 26756 (22410 unique)
- *Vulnerable Dependencies:* 5
- *Vulnerabilities Found:* 13
- *Vulnerabilities Suppressed:* 0
- ...

Summary

Display: [Showing Vulnerable Dependencies \(click to show all\)](#)

| Dependency | Vulnerability IDs | Package | Highest Severity | CVE Count | Confidence | Evidence Count |
|-------------------------------|-------------------|--|------------------|-----------|------------|----------------|
| Cesium.js | | pkg:javascript/DOMPurify@3.0.6 | HIGH | 3 | | 3 |
| Cesium.js | | pkg:javascript/DOMPurify@3.0.6 | HIGH | 3 | | 3 |
| express 5.1.0 | | pkg:npm/express@5.1.0 | MEDIUM | 1 | | 7 |
| index.js | | pkg:javascript/DOMPurify@3.0.6 | HIGH | 3 | | 3 |
| index.js | | pkg:javascript/DOMPurify@3.0.6 | HIGH | 3 | | 3 |

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

26 brukerhistorier sprint 1

Sprint 1

| Brukerhistorie | Prioritet | Status (antall akseptansekriterier fullført) | Kommentar |
|--------------------------------|-----------|---|--|
| Finn tilfluktsrom | HØY | 2/2 | |
| Registrer bruker | HØY | 4/5 | Mangelende personvernserklæring |
| Logg inn | HØY | 5/5 | |
| Se beredskapslager | HØY | 3/3 | |
| Oppdater beredskapslager | HØY | 3/3 | Brukerflyten for å oppdatere varer ble endret etter sprint 1, men kravet ble oppfylt |
| Beredskapsgrad | HØY | 3/3 | Endrede akseptansekriterier i tråd av PO-enes ønsker. |
| Refleksjon etter krise | LAV | 2/3 | Implementert men rom for utbedring til neste sprint |
| Admin logg inn | HØY | 3/10 | Manglet super-adminbruker, 2FA, og mekanismer for forhindring av bruteforce. Fikk til kryptering av passord, lage sikkert passord med tilbakemeldinger |
| Kart ikoner | HØY | 0/7 | Fikk ikke tid |
| Kartoppdatering | HØY | 0/10 | Fikk ikke tid |
| Bli med eller opprett husstand | HØY | 0/2 | Fikk ikke tid. Brukte mock husstand under utvikling |

27 brukerhistorier sprint 2

Sprint 2

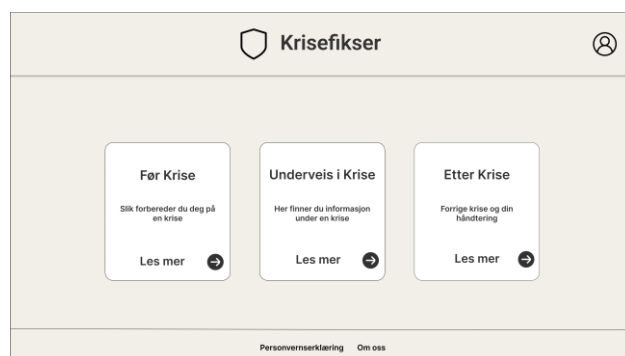
| Brukerhistorie | Prioritet | Status | Kommentar |
|---|-----------|---------------------------------------|---|
| | | (antall akseptansekriterier fullført) | |
| Registrer bruker (Resterende fra Sprint 1) | HØY | 5/5 | Personvernserklæring implementert |
| Refleksjon etter krise (Resterende fra Sprint 1) | HØY | 3/3 | Mulighet for å dele refleksjonsnotat med andre brukere |
| Adminlogg inn | HØY | 3/10 | Ikke gjort noen endringer i denne sprinten |
| Kart ikoner (Resterende fra Sprint 1) | HØY | 7/7 | |
| Kartoppdateringer (Resterende fra Sprint 1) | HØY | 9/10 | Rakk ikke implementere varslinger for brukere i nærheten av episenteret |
| Bli med eller opprett husstand (Resterende fra Sprint 1) | HØY | 2/2 | |
| Se kriseberørt område som ikke-registrert bruker | HØY | 3/4 | Fikk ikke tid til å implementere varslinger for hendelser som inntreffer i nærheten av brukere |
| Generell info | MIDDELS | 2/2 | |
| Personvern | MIDDELS | 1/1 | |
| Om oss | LAV | 2/2 | |
| Beredskapslagerkontrollvarsling | MIDDELS | 2/3 | Fikk ikke tid til å implementere vibrasjon eller lyd |
| Se-hjem-i-kart | MIDDELS | 3/3 | |
| Invitere til husstand | HØY | 2/2 | |
| Bytte husstand | LAV | 3/5 | Brukere kan tildeles ny husstand, dette gjøres gjennom invitasjon. En bruker kan kun være tildelt én husstand |
| Gamification | LAV | 1/1 | Implementert med en quiz |

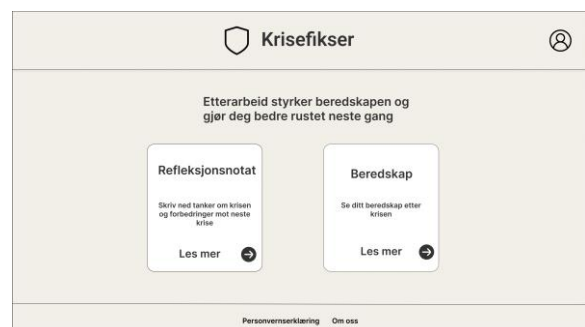
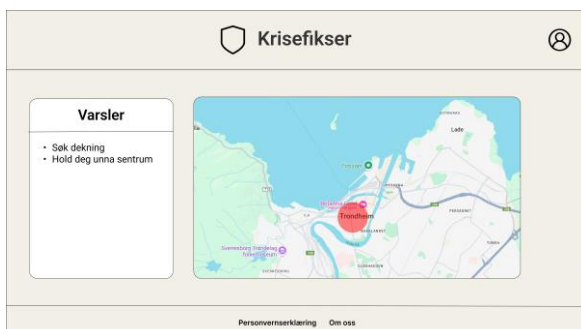
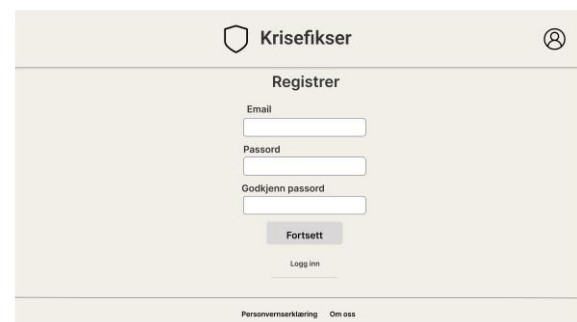
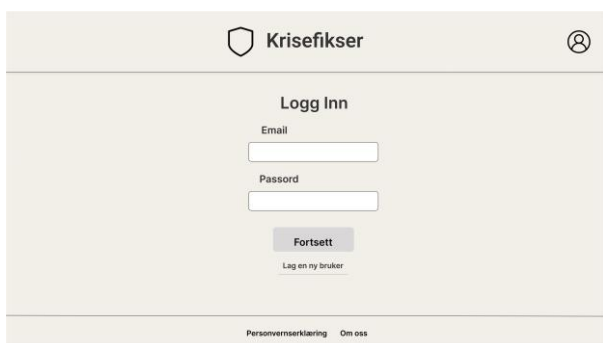
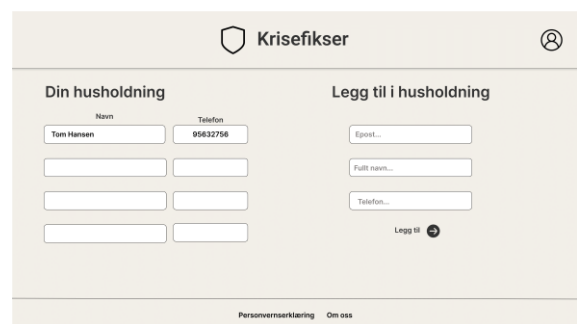
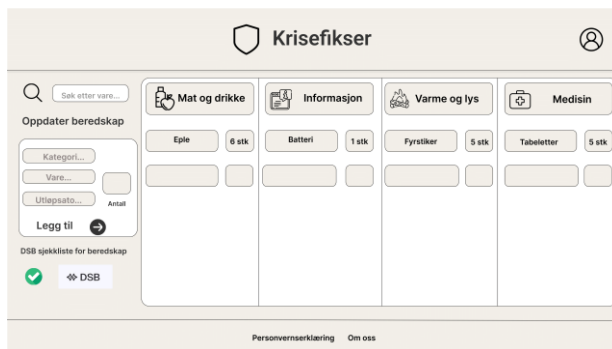
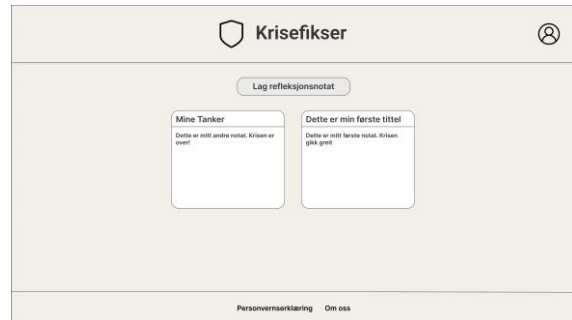
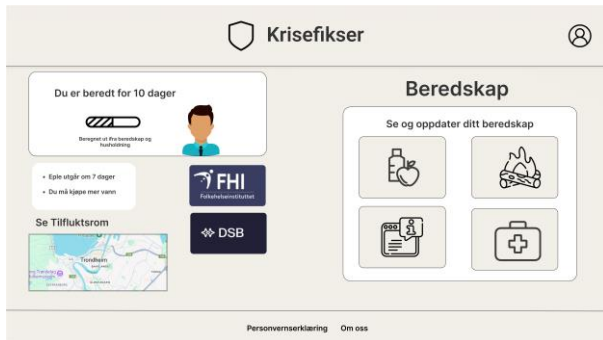
28 ikke fullførte brukerhistorier

Brukerhistorier som ikke bli inkludert i produktkø

| Brukerhistorie | Prioritet | Kommentar |
|----------------------------------|-----------|------------------------------|
| Antall i husstand | MIDDELS | Nedprioritert |
| Les nyheter | LAV | Nedprioritert |
| Finn møteplass | MIDDELS | Nedprioritert |
| Opprette beredskapsgruppe | MIDDELS | Nedprioritert av PO |
| Legge til beredskapsgruppe | MIDDELS | Nedprioritert av PO |
| Delt beredskapslager | MIDDELS | Nedprioritert av PO |
| Se posisjon til andre | MIDDELS | Nedprioritert av PO-by-proxy |
| Glemt passord | MIDDELS | Nedprioritert av PO-by-proxy |
| Få varslng om å skru på posisjon | LAV | Gikk i glemmeboka |
| Scenario | LAV | Nedprioritert |
| Gamification Admin | LAV | Nedprioritert |
| Invitere Admin | MIDDELS | Nedprioritert av PO-by-proxy |

29 Wireframes





30 Domenemodell

