

# Software Requirements Specification

Visning av Eiendomsverdi for brukere hos Bulder Bank

Versjon 0.1

Prepared by Markus Nordstrønen

## 1. Introduksjon

### 1.1 Formål

Denne kravspesifikasjonen beskriver målet med løsningen, samt hvilke krav som må oppfylles for at løsningen skal fungere etter intensjonen.

### 1.2 Scope

I forbindelse med markedsføring får Bulder god tilbakemelding på å la brukerne sjekke sin boligverdi i Bulder sin lånesøknad. Utfordringen vår i dag er at denne løsningen krever en ganske tung pålogging, budskapet på landingssidene er veldig rettet mot at kunden skal søke lån, og alle som skal sjekke boligverdi ødelegger vår evne til å måle om lånesøknaden fungerer skikkelig.

Bulder ønsker derfor å implementere en frittstående løsning der brukere hos Bulder Bank kan se sin boligverdi. Brukeren logger på med BankID, løsningen viser brukerens boliger, og tilhørende verdi. I tillegg presenteres brukeren med en lenke til lånesøknaden til Bulder.

Presiseringer og begrensninger:

- I endelig løsning må brukerne være identifisert gjennom BankID, men det er ikke i scope for denne oppgaven.
- Brukeren må ikke være kunde i Bulder for å bruke løsningen.

### 1.3 Referanser

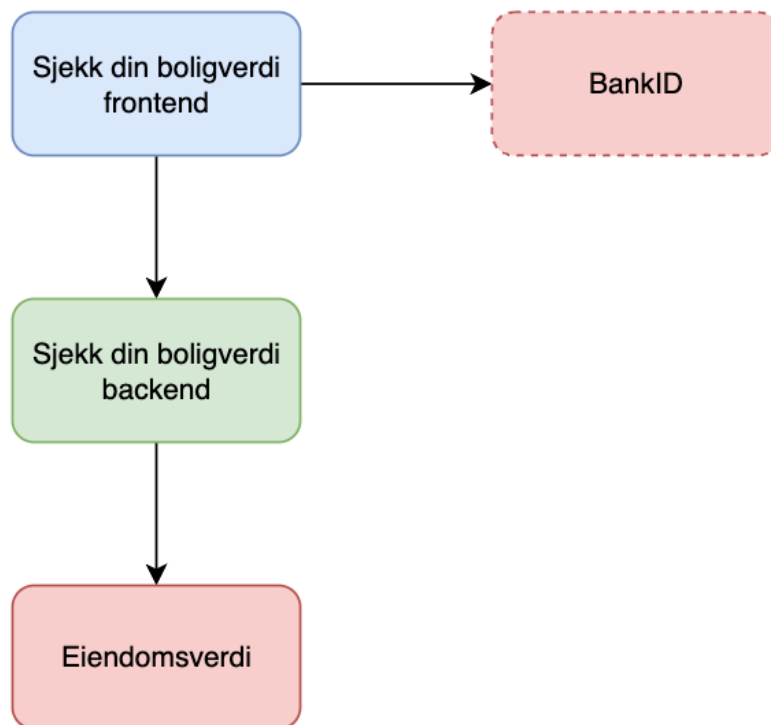
- API Spesifikasjon for Eiendomsverdi
- API Spesifikasjon for BankID (OpenID Connect)
- Designskisse

## 2. Produktoversikt

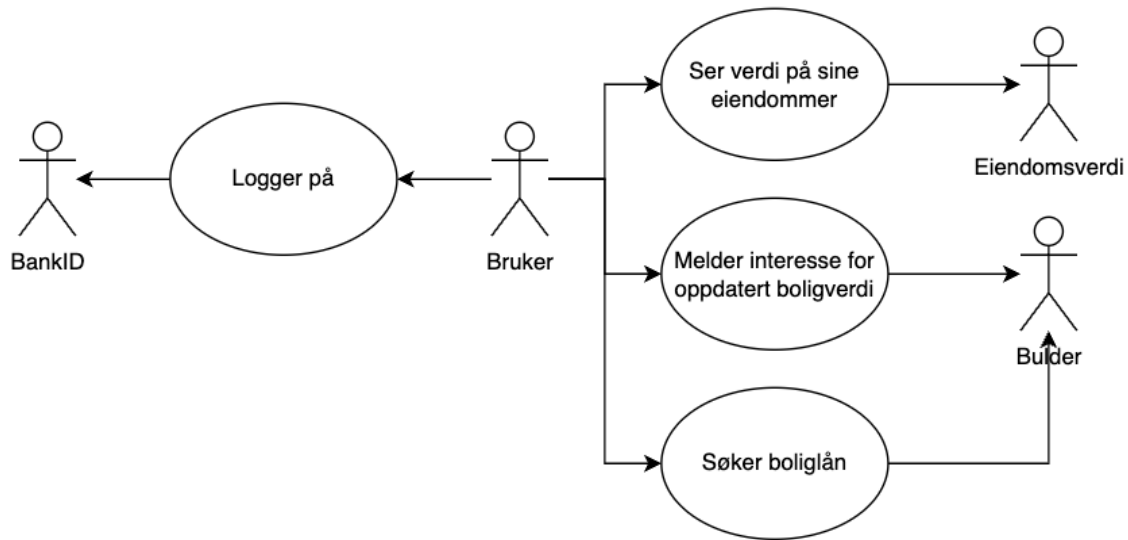
## 2.1 Perspektiv

Løsningen implementeres som en standalone nettside med en backend som henter og filtrerer data fra eiendomsverdi. Det er viktig at tilgangsnøkler til Eiendomsverdi håndteres sikkert på backend og at backend ikke lekket data til frontend ut over det som er nødvendig for å sikre brukeropplevelsen.

Overordnet applikasjonsarkitektur og eksterne integrasjoner:



## 2.2 Funksjoner og aktører



## 2.3 Begrensninger

- BankID er ikke i scope for første iterasjon.
- Løsningen må skrives for web, og helst bruke React som rammeverk.
- Backend webserver må kunne pakkes som en dockercontainer og deploys på kubernetes.
- Løsningen integrerer mot eiendomsverdi sine nye REST APIer.

## 2.4 Brukere

Brukere er alle som ønsker å sjekke sin boligverdi for så å kunne vurdere å flytte sitt lån til Bulder Bank. Typisk vil vi promotere løsningen i reklame eller i SoMe kampanjer.

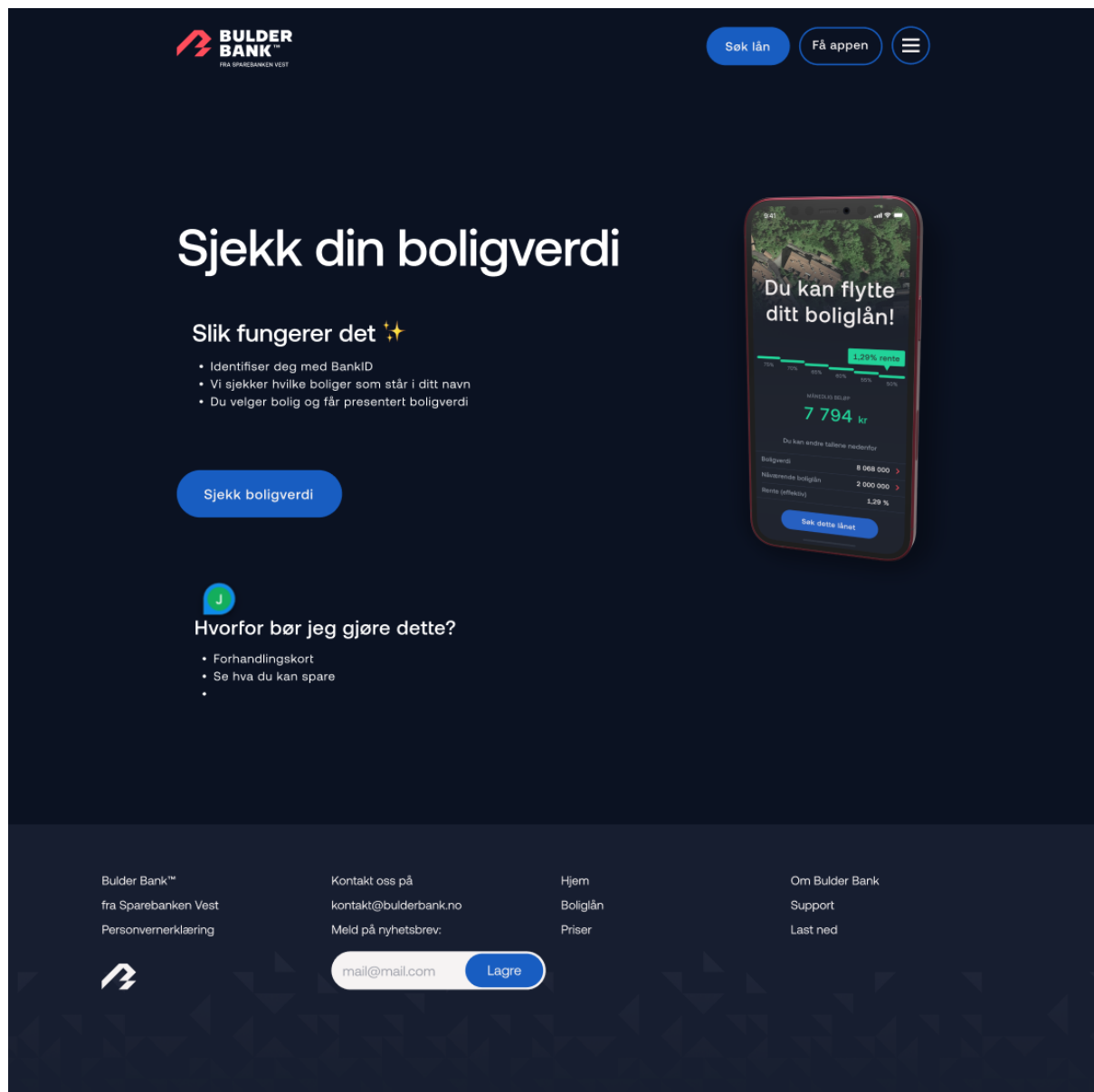
## 2.5 Antagelser og avhengigheter

- REST api fra Eiendomsverdi er ikke ferdig, men skal kunne tilby estimer i løpet av Oktober. Vi venter på tilgang til REST api fra Eiendomsverdi.

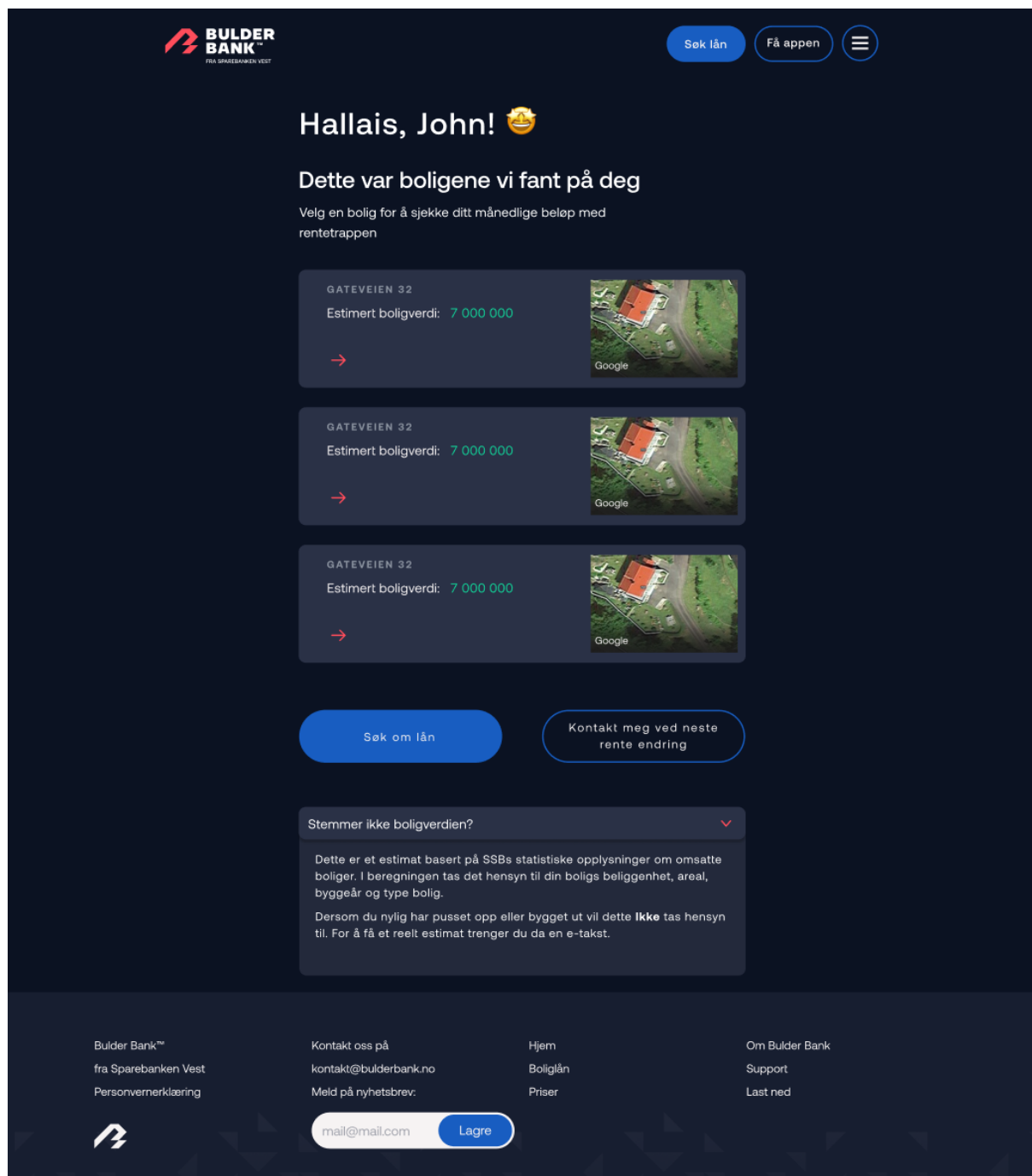
## 3. Krav

### 3.1 UX skisser

#### 3.1.1 Landingsside



### 3.1.2 Boligverdi



### 3.2 Funksjonelle krav

- F01 - Brukeren skal få en oversikt over sine boliger og estimert verdi på dem.
- F02 - Brukeren skal få opp kartvisning for boligen. Her kan dere benytte google maps.
- F03 - Brukeren skal kunne legge igjen epost for å bli varslet ved neste endring i boligverdi.
- F04 - Brukeren skal få mulighet til å hoppe over til Bulder sin låneflyt ("Søk om lån").

F05 - Dersom vi ikke finner noen boliger på brukeren skal hun/han få informasjon om hva som kan være galt.

F06 - Bulder skal kunne se bruken av tjenesten og hvordan brukerne benytter seg av den. Bulder bruker i dag Mixpanel for denne typen flytanalyse.

### 3.3 Ikke-funksjonelle krav

#### 3.3.1 Ytelse

NF01 - Brukeren skal få opp sin oversikt innen 5 sekunder. I mellomtiden skal det vises en spinner.

#### 3.3.2 Sikkerhet

NF02 - Brukeren skal kun få opp eiendommer der hun/han er hjemmelshaver. Det skal ikke være mulig å slå opp eiendommer for andre personer.

NF03 - Det skal kun presenteres data som er relevant for visningen. Dvs. skal det ikke lekke utilsiktet data fra API eller backend til frontend.

NF10 - Tilgangsnøkler for API skal lagres i en passende secret manager. Ikke i kode eller config!

NF19 - Løsningen bør testes mot OWASP topp 10. <https://owasp.org/Top10/>

#### 3.3.3 Stabilitet og robusthet

NF04 - Løsningen skal kunne støtte 200 samtidige brukere uten å gå under kravet til NF01.

NF13 - Løsningen skal ha 99,9% oppetid (maks 45 min nedetid per måned).

NF05 - Backend skal kjøre på to instanser (pods) og skal være stateless.

NF06 - Løsningen bør ha dashboards for overvåking av metrikker (antall brukere, status på oppslag (feil, ok, antall boliger).

NF07 - Løsningen bør ha alarmer dersom feilraten er veldig høy, eller API mot eiendomsverdi er utilgjengelig.

NF18 - Dersom Eiendomsverdi er nede må det vises en feilmelding til brukeren.

#### 3.3.4 UU/Tilgjengelighet

NF08 - Løsningen skal være tilgjengelig også for svaksynte og blinde. Score minimum x i y verktøy.

NF17 - Løsningen skal følge WCAG standarden.

### 3.5 Applikasjonsdesign og implementasjon

#### 3.5.1 Installasjon

NF09 - Applikasjonen skal kunne bygges som et docker-image og produsjonssettes på Kubernetes.

### 3.5.2 Vedlikeholdbarhet

NF11 - Frontend bør implementeres i React.

NF12 - Backend kan implementeres i C# (.NET) eller Go, men det er ikke et absolutt krav.

NF14 - Kode skal ligge i github.

### 3.5.4 Gjenbruk

NF15 - Applikasjonen skal kunne integreres i bulderbank.no. Dere får en branch av bulderbank.no for test.

### 3.5.6 Kostnad

NF16 - Applikasjonen bør optimaliseres for å minimere kostnad.

## 4. Godkjenning

Produktleveransen godkjennes når funksjonalitet og kvalitetskrav er oppfylt i henhold til denne spesifikasjonen. Løsningen presenteres som demo og vurderes av kravstiller.

## 5. Appendixes

### 5.1 API dokumentasjon Eiendomsverdi