

Kravdokument

Iver Sande, Mikal Bø

Mai 2024



Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
16.02.24	1.0	Startpunkt	Mikal Bø og Iver Sande
06.05.24	1.1	Utbedret forklaringer	Iver Sande
12.05.24	2.0	Avsluttet rapport	Mikal Bø og Iver Sande

Innhold

1	Innledning	3
2	Funksjonalitet	4
2.1	Opprette Bruker	4
2.2	Endre Bruker	6
2.3	Slette Bruker	7
2.4	Oversikt Brukere	8
3	Modeller	10



1 Innledning

I dette dokumentet vil vi beskrive hvordan prosjektet fungerer. Det starter med en oversikt over funksjonene til systemet. Disse vil bli beskrevet med brukerhistorier og brukstilfelle diagrammer. Det vil også sies litt om kravene som eksisterer for hver av brukstilfellene.

Deretter vil det vises til noen modeller som modelerer systemet for å gjøre det tydelig hva som eksisterer av funksjonalitet og hvor denne funksjonaliteten ligger i systemet.



2 Funksjonalitet

2.1 Opprette Bruker

Som Portalbruker

Ønsker jeg å opprette en bruker

Slik at kunden kan logge inn på sin bruker

Funksjonelle krav :

Lager ikke brukere som eksisterer

Felt for all data som skal lagres i brukeren og trenger input

Tilbakemelding om du har gjort noe feil

Ikke Funksjonelle krav :

Lett å navigere frem til lage bruker

Feilhåndtering hvis noe ikke funker

Det skal sjekkes at ting er riktig fylt inn slik at det er vanskelig å gjøre feil

Scenario :

Gitt at all informasjon er fylt ut riktig

Når jeg trykker lag ny bruker

Så skal brukeren opprettes hvis den ikke finnes fra før

Og den blir da tilgjengelig under listen med brukere

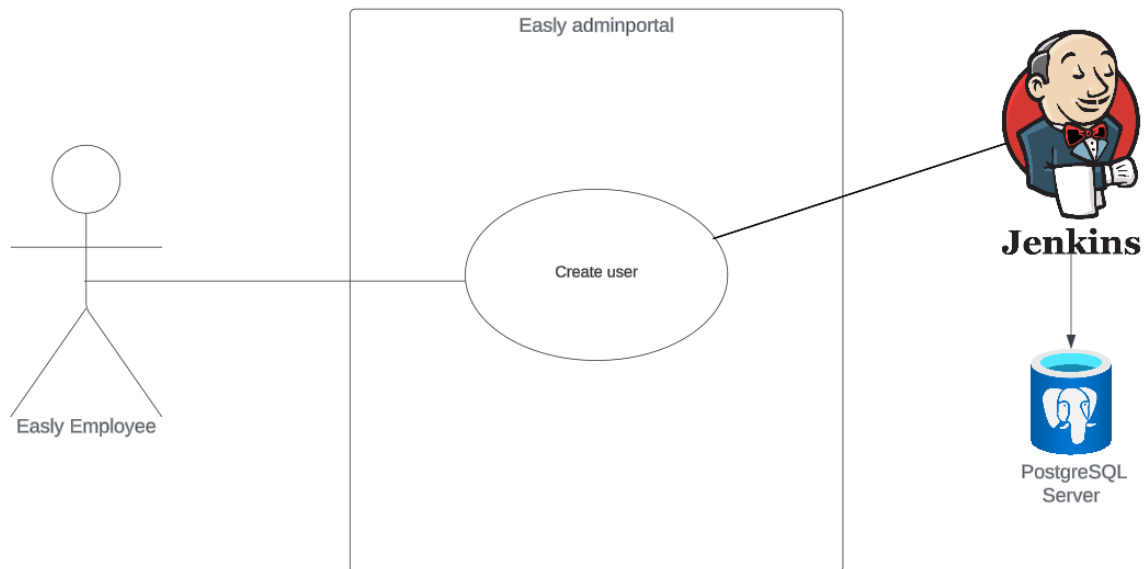
Scenario :

Gitt at informasjonen er fylt ut feil

Når jeg trykker lag ny bruker

Så skal brukeren ikke opprettes

Og du skal få en tilbakemelding på hvilke felter som er fylt ut feil



Figur 2.1: Lag Bruker Use Case

2.2 Endre Bruker

Som Portalbruker

Ønsker jeg å endre brukeren

Slik at vi har riktige data

Og slik at oversikten for selskapet stemmer

Funksjonelle krav :

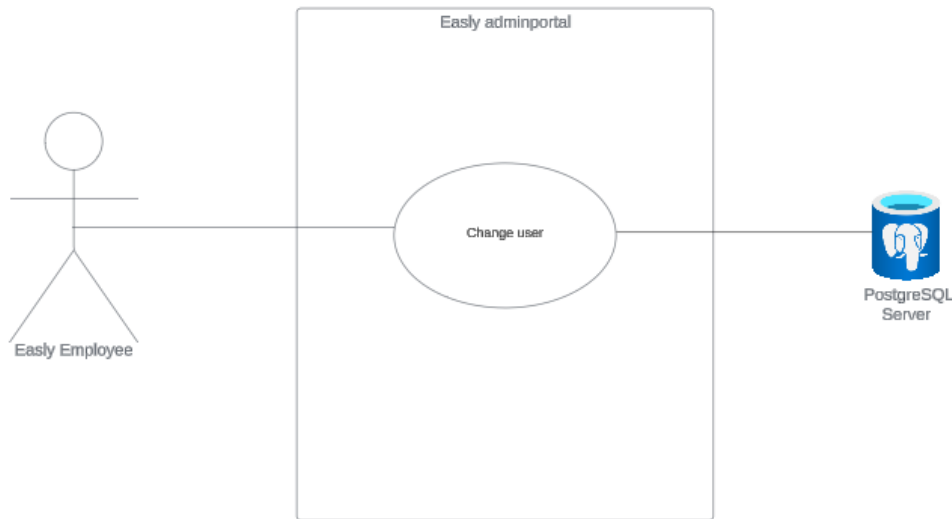
Tilbakemelding om du har gjort noe feil

Ikke Funksjonelle krav :

Ikke mulig å gjøre endringer slik at unike felt blir overlappende med andre brukere

Scenario :

Gitt at du skal endre en bruker Når du trykker endre bruker Så skal du få en full oversikt over alle feltene som kan endres Og ved inputs som leder til ikke unike felter skal du få en feilmelding Og ved godkjente inputs skal du få en tilbakemelding på at den er blitt endret



Figur 2.2: Endre Bruker Use Case

2.3 Slette Bruker

Som Brukerstøtte

Ønsker jeg å slette brukeren

Slik at vi får bedre oversikt over aktive brukere

slik at ledelsen fakturerer kunden korrekt

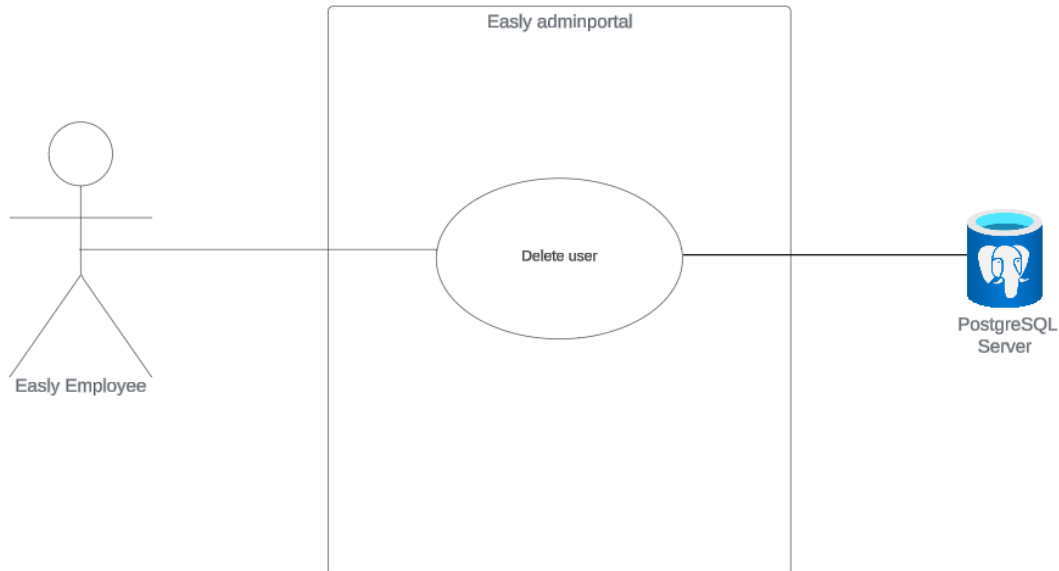
Ikke funksjonelle krav :

Lett å navigere frem til å slette bruker

vanskelig å slette ved uhell

Scenario :

Gitt at jeg vil slette en bruker



Figur 2.3: Slett Bruker Use Case

Når jeg trykker på slett

Så kommer det en bekreftelse

Og den krever enda en handling før du faktisk godkjenner

2.4 Oversikt Brukere

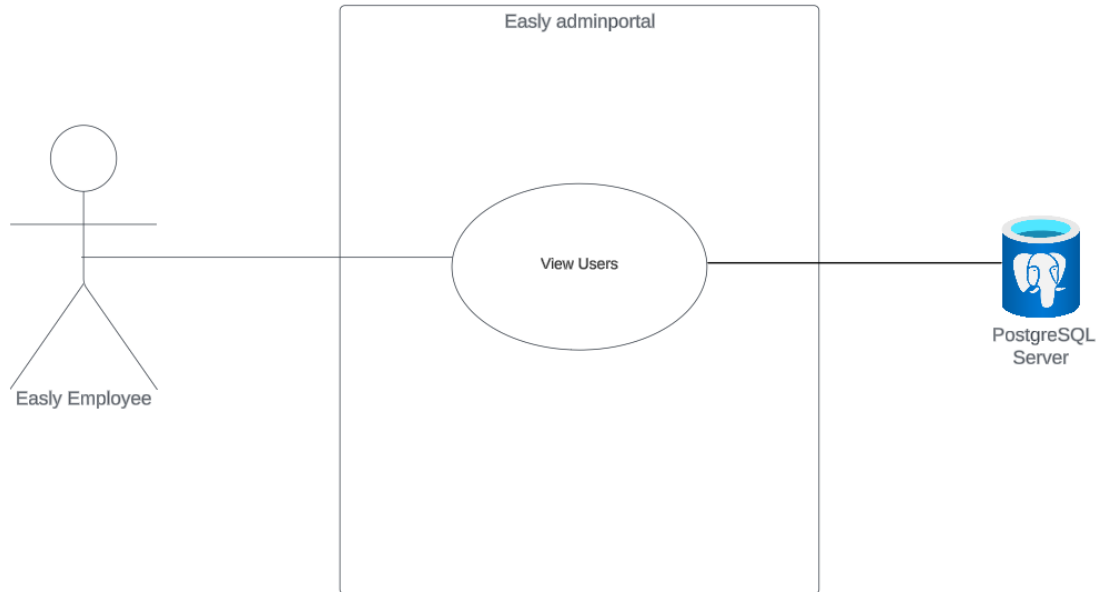
Som daglig leder

Ønsker jeg en oversikt

Slik at jeg lett kan oppdatere styre om antall kunder.

Ikke funksjonelle krav :

God oversikt over alle brukere



Figur 2.4: Se Brukere Use Case

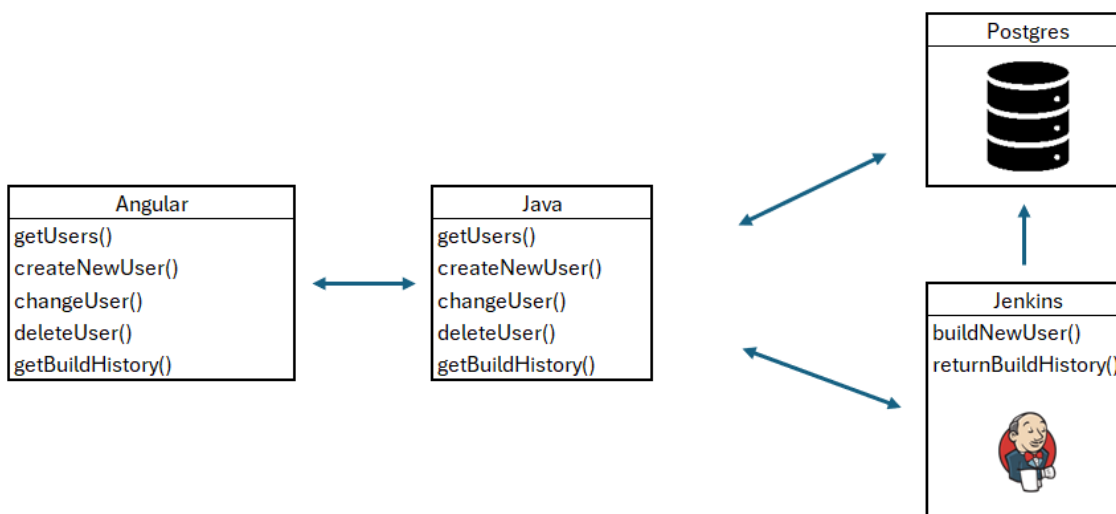
Funksjonelle krav :

Alle brukerne skal vises

3 Modeller

Overordnet systemflyt

Domenemodellen 3.1 viser flyten for systemet i grove trekk. Denne modellen viser funksjonene i hver del av systemet med beskrivende navn og hvordan systemene kommuniserer. Kommunikasjonen foregår der det er piler og i begge retninger der det er piler som peker på begge modulene. En ting som kan nevnes om brukerne er at Jenkins kun har ansvar for å lage nye brukere, all annen brukermanipulasjon skjer gjennom databasen direkte.

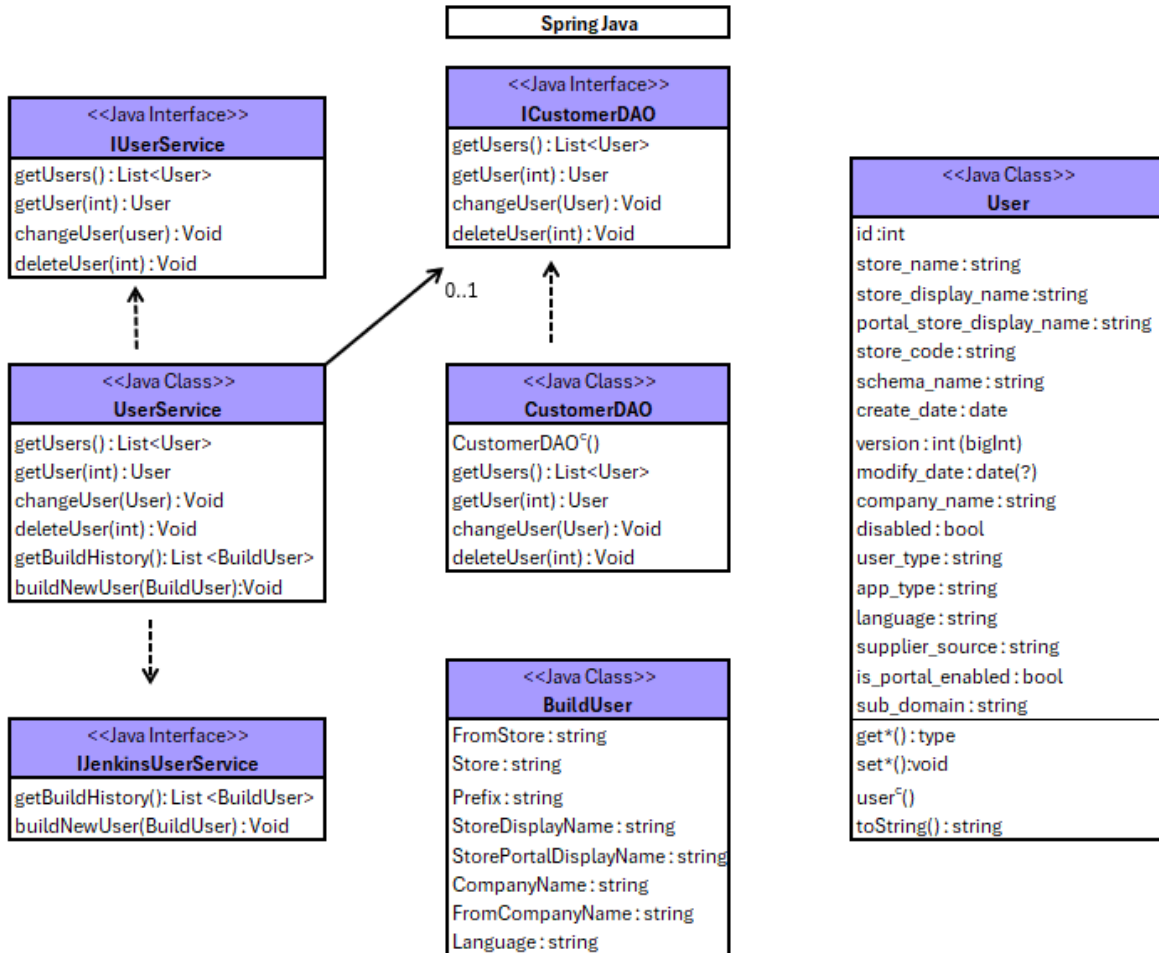


Figur 3.1: Domenemodell



Klassediagram for Spring backend

Klassediagrammet her viser hvordan backend servicen er bygget opp. Denne strukturen var den planlagte før det ble startet, men fikk noen endringer i løpet av utviklingsfasen.



Figur 3.2: Klassediagram