

# Kravspesifikasjon

## Reservasjonssystem for parkeringsplasser

Gruppe 12:  
Jonas Eliassen  
Mats Engelen  
Lars Erik Faber  
Håkon Marthinsen  
Charis Schille

# Innholdsfortegnelse

<b>1</b>	<b>Problemstilling</b>	<b>4</b>
<b>2</b>	<b>Løsningen</b>	<b>4</b>
<b>3</b>	<b>Begrensninger og antagelser</b>	<b>4</b>
3.1	Begrensninger . . . . .	4
3.2	Antagelser . . . . .	4
<b>4</b>	<b>Ansvarsområde og Avhengigheter</b>	<b>5</b>
<b>5</b>	<b>Funksjoner i applikasjonen</b>	<b>5</b>
<b>6</b>	<b>Begreper og definisjoner</b>	<b>6</b>
<b>7</b>	<b>Brukerklasser</b>	<b>6</b>
7.1	Bruker . . . . .	6
7.2	Bedrift . . . . .	7
7.3	Administrator . . . . .	7
<b>8</b>	<b>Use Case diagrammer</b>	<b>7</b>
8.1	Use Case - Bruker (Leietaker) . . . . .	7
8.2	Use Case - Bedrift . . . . .	8
8.3	Use Case - Administrator . . . . .	8
<b>9</b>	<b>Personaer</b>	<b>9</b>
9.1	Jørgen Moe . . . . .	9
9.2	Sigurd Olson . . . . .	9
9.3	Karina Gausrud . . . . .	10
9.4	Karin . . . . .	11
9.5	Jørn Berg . . . . .	12
<b>10</b>	<b>Funksjonelle krav</b>	<b>12</b>
10.1	Kontoopprettelse . . . . .	12
10.2	Innlogging . . . . .	13
10.3	Verifiseringskode . . . . .	13
10.4	Presentasjon av parkeringsplasser (Feed) . . . . .	14
10.5	Leie parkeringsplass . . . . .	14
10.6	Tilbakemelding på innlegg (Anmeldelser) . . . . .	15
10.7	Opprette parkeringsplasser . . . . .	15
10.8	Leiebetingelser . . . . .	16
10.9	Betaling . . . . .	16
10.10	Administrering . . . . .	16
10.11	Persistent data . . . . .	18
<b>11</b>	<b>Ikke-funksjonelle krav</b>	<b>18</b>
11.1	Tilgjengelighet . . . . .	18
11.2	Sikkerhetskopi . . . . .	19
11.3	Tredjepart . . . . .	19
11.4	Brukerdata og sikkerhet . . . . .	19
11.5	Programvare . . . . .	19
11.6	Samsvar . . . . .	20
11.7	Avhengigheter . . . . .	20

<b>12</b>	<b>Prototype</b>	<b>21</b>
12.1	Sammendrag . . . . .	21
12.2	UML . . . . .	21
12.3	Innstallasjon . . . . .	22
12.4	Bruksmanual . . . . .	23
12.5	Kjøre tester . . . . .	23
12.6	Implementerte krav med tester . . . . .	24
12.7	Avhengigheter . . . . .	25
12.8	Begrensninger . . . . .	25
<b>13</b>	<b>Diagrammer</b>	<b>26</b>
13.1	Sekvensdiagram - Leie en parkeringsplass. . . . .	26
13.2	Dataflytdiagram - Tilbakemelding . . . . .	27
13.3	Sekvensdiagram - Opprette reservasjon . . . . .	27
13.4	Tilstandsdiagram - Registrere parkeringsplass . . . . .	28
13.5	Aktivitetsdiagram - Leie en parkeringsplass . . . . .	29
13.6	Sekvensdiagram - Leie ut en parkeringsplass . . . . .	30
13.7	Dataflytdiagram - Opprette parkeringsplass . . . . .	30
13.8	Dataflytdiagram - Slette Parkeringsplass . . . . .	31
13.9	Aktivitetsdiagram - Legge til medlem i organisasjon . . . . .	32
13.10	Sekvensdiagram - Oppdatere Parkeringsplass . . . . .	33
<b>14</b>	<b>Estimering</b>	<b>33</b>
14.1	Størrelser . . . . .	33
14.2	Estimeringstabell - Funksjonelle krav . . . . .	34
14.3	Estimeringstabell - Ikke funksjonelle krav . . . . .	36

# 1 Problemstilling

Ofte finner man ikke parkeringsplass mens man er i farta. Det er tett på senterene, gateparkering er privat eller ulovlig og man har ofte dårlig tid. Gjerne etter litt leting finner man en parkering 500 meter unna, og når man kommer dit er det selvfølgelig fullt. Hvordan kan man unngå å miste så mye tid, og i tillegg finne en parkeringsplass?

## 2 Løsningen

Tjenesten baserer seg på delingsøkonomi og enheten som deles er parkeringsplasser. Ideen er at bedrifter og enkeltpersoner kan registrere en eller flere parkeringsplasser de eier for utleie til enkeltpersoner. Dermed kan disse utleierene tjene penger passivt, samt gjøre det enklere for privatpersoner å forsørge parkering. Parkeringsplassen de legger ut vil være merket med bredde, lengde og høyde, samt om den er reservert for el-biler eller handicappede. På denne måten kan alle som bruker tjenesten få et godt innblikk i parkerings-bildet.

Bruker logger inn som en bedrift eller en vanlig bruker. Brukere kan redigere og legge ut sine registrerte parkeringsplasser, samt leie parkeringsplasser fra andre brukere. Bedrifter kan kun legge ut parkeringsplasser, men har muligheten til å legge til brukere som medlemmer av bedriften. Dette gir de spesialtillatelse til parkeringsplasser som er reservert kun for medlemmer av en spesifikk bedrift. Medlemmer av bedriften vil også kunne legge ut og merke parkeringsplasser for andre bedrifter.

Tjenesten er bygget opp slik at alle brukere har tilgang til en feed som viser en oppstilling av de mest relevante parkeringsplassene, en side for registrering av parkeringsplasser og en profilside. For å kunne reservere en parkeringsplass, må bruker oppgi informasjon om bilen de benytter. På feeden rangeres de tilgjengelige parkeringsplassene etter avstand til bruker eller et spesifikt område som bruker spesifiserer, og vises kun dersom de er passende med bilen. Bruker kan også trykke seg inn på et spesifikt innlegg for å se mer detaljer om parkeringsplassen som leies ut, og kan legge en anmeldelse på parkeringsplasser de har brukt. På denne måten får utleiere tilbakemelding på hva de gjør riktig og hva de gjør galt, som bidrar til å øke engasjementet til å gjøre en god jobb.

Kunde av tjenesten vil kunne opprette administratorbrukere som har spesielle egenskaper for administrering av feeden og brukerne. Dette vil være nødvendig for å minke upassende/ulovelig innhold og øke brukeropplevelsen.

## 3 Begrensninger og antagelser

### 3.1 Begrensninger

**Betaling :** På grunn av lovverk så kan ikke vi implementere vår egen betalingssystem og vil trenge en tredjepart

**Servere rundt i landet :** På grunn av ansvarsområde så har vi ikke kapasitet til å sette opp våre egne sever og må dermed leie av en tredjepart.

**Cookies :** På grunn av lovverk og at en bruker skal kunne få tilgang til applikasjonen må de samtykke cookies.

**Googlemaps API :** På grunn av begrenset tid og ressurser så er vi ikke stand til å lage vårt eget kart API.

### 3.2 Antagelser

**Opprette en parkeringsplass i applikasjonen :** Når en bruker skal opprette en parkeringsplass for første gangen eller for alle andre ganger, så skal det være så enkelt og kjapt som mulig, at de

ikke får følesen av at de kaster bort tid (og penger). Det skal være så enkelt at når en bruker har logget seg inn så skal det være en knapp der det står opprett en parkeringsplass. Der skal de kunne fylle in informasjon om, hvor mange etasjer det er, hvor stort område er, hvor ligger den og hva slags parkeringsregler har den for område. Når eieren for parkeringsplassen har opprette starts delen for parkeringsplassområdet, så er det viktig at de kan fylle innholdet på parkeringsplassen, som hvor mange parkeringsplasser det i hver etasjer, hva slags parkeringsplass er det (er plassen handicappede, elektrisk, motorsykkel osv.), skal parkeringsplassen ha en bestemt ID og time prisen for å parker hos dem. Imens eieren for parkeringsplassen oppretter den så er det viktig at systemt lagrer / tar backups av det de gjør i tilfelle noe skulle skje. Når de er ferdig med å opprette en parkeringsplass, så skal det være så enkelt at de kan si at den er åpen for bruk og alt etter dette punkte her er automatisk, når det gjelder å leie fra dem.

**Oppdater en parkeringsplass i applikasjonen :** Eieren av en parkeringsplass vil kanskje endre på noe i den fysiske oppsette av parkeringsplassen eller det som ligger ut i applikasjonen om det generelle informasjonen, om det er å legge til en ekstra etasje, flere/fære parkeringsplasser, endring av strukturen av parkeringsplassen, informasjonsfeil. De skal finne en knapp vedsiden av parkeringsplassen som de eier på applikasjonen i minne parkeringsplassener. Når de trykker på rediger parkeringsplass, så skal de få muligheten til å rediger/endre på informasjonen som er lagret i den parkeringsplassen. Når eieren er ferdig med å rediger/endre så skal de kunne trykke lagre og da vil systemet gjøre resten.

**Leie :** Når en bruker trenger et sted å parkere, så skal det være så enkelt som å logge seg på og finne knappen lei en parkeringsplass. Når brukeren har trykket på lei en parkeringsplass så vil de bli ført til en side der de kan søke etter bestemte parkeringsplasser etter dems behov, om det er handicappede parkering, elektrisk parkering eller om det er en normal parkeringsplass. De har også muligheten for å bestemme hvor de skal parker om det er på dems vanlige parkeringsplass eller om det er på et nytt område. Når de har valgt hvor de skal parker og oppfylt alt av informasjon som trengs så skal de kunne betale for det, hvis de har valgt tid de skal stå der (fra og til tid). Da vil applikasjonen kalkulere prisen for tiden bruken skal stå der (i time pris) og da skal de kunne betale for det. Etter de har trykket betal så skal de få en melding om at det er godkjent og en kvitering for at de har betalt for den.

**Administrator og administrer :** Det er viktig som en administrator at alt som er applikasjonen som vises fra til de andre brukeren stemmer. At ingeng lager flaske brukere og oppretter falske parkeringsplasser som kan bli brukt til svindel eller stjeling av andres privatinformasjon. Det kan hende at man må slette disse brukerne eller banlyse dem og det samme gjelder for de falske parkeringsplassen.

## 4 Ansvarsområde og Avhengigheter

Tjenesten skal dekke områder som framvisning, utleie og reservasjon av parkeringsplasser, brukerinntillinger, lagring av brukerdata gjennom cookies, lagring av parkeringsplasser, lagring av brukeranmeldelser. Tjenesten skal ikke håndtere transaksjoner, men etterspør bank. GPS koordinater håndteres av en tredjepart. Tjenesten har heller ikke ansvar for hosting.

Applikasjonen bygger på automatiseringsverktøy Maven med JDK versjon 11. Jackson brukes for datalagring for datalagring og Vue som er et åpent visningsrammeverk og brukes for visning av tjenesten. Javalin er web-rammeverket som brukes, og testene kjøres med JUnit.

*Avhengigheter er listet under ikke-funksjonelle krav.*

## 5 Funksjoner i applikasjonen

Programmet henter inn info om brukere, poster, parkeringsplasser og reservasjoner fra en database (i dette tilfellet json-filer). Det programmet gjør nå er å hente inn data fra controller til javalin som

generer front-end for oss. Hver side henter da ut nødvendig info hva hver.json fil slik at nødvendig informasjon som adresse og ledighet til leie vises.

**Bruker** - Det er mulig å vise oversikt over parkeringsplasser, trykke seg inn til en parkeringsplass, og å legge til en reservasjon, brukeren kan også se oversikt over p-plassene som den har reservert. Det er mulig å legge ut en ny parkeringsplass som en utleier. Det er mulig å vise bilder av parkeringsplassen via et link system, da opplastning direkte til siden ikke er lagt inn.

**Corporation bruker** - kan kun legge ut parkeringsplasser. Dette gjør man ved å gå til “My parking spots” hvor “Create new parkingspot” står øverst. For å lage et nytt parkingeringsobjekt må man trykke på denne knappen og fylle ut all nødvendig info(feilmelding dersom ikke alt har noen verdi), og man kan i tillegg velge ved bruk av ikoner dersom det er handikappet eller el-bil plass. Corporation skal kunne få betalt for å leie ut p-plasser.

**Admin** - Denne brukeren kan suspendere eller slette/banlyse brukere, slette poster og parkeringsplasser. Dette er en bruker-type som i senere iterasjoner kan gi rettigheter til andre brukere (f.eks. modereringsbrukere).

## 6 Begreper og definisjoner

Begrep	Beskrivelse
Applikasjon	Et programvare som benytter datamaskinens ressurser til en oppgave som brukeren ønsker utført
Bruker	Besøkende som samhandler med systemet
Brukerkonto	En som kan kun leie parkeringsplasser
Bedriftskonto	En som kan kun leie ut parkeringsplasser til andre
Administrator	Brukerkonto med administratorrettigheter
Feed	Opplisting av innlegg
Tjenesten	Tjenesten denne dokumentasjonen omhandler
Systemet	Det underliggende systemet til tjenesten
Personas	Er et annet ord for rollefigur, enkeltmenneske, karakterpersonlighet
ID	Identifikator
FK	Funksjonelt Krav (brukes som ID)
IFK	Ikke-Funksjonelt Krav (brukes som ID)
Innlegg	Brukerinnlegg i systemet (post) som består av en eller flere parkeringsplasser
Produktet/ Tjenesten	Applikasjonen som dokumenteres
Kunde	Gruppe som eier produktet
Push-notifikasjoner	Varsler på desktop eller mobil
Cookies	Informasjonskapsler
TOS	Terms of Service, liste over bruksrettigheter og bruksområde for brukerdatabaser
EZpark	Kodeord for applikasjonen som dokumenteres

## 7 Brukerklasser

En oversikt over roller og egenskaper for de forskjellige brukerklassene.

### 7.1 Bruker

Brukerkontoen er en standardkonto som hvem som helst kan opprette. Som en bruker, kan man registrere sine egne parkeringsplasser og leie andre brukeres parkeringsplasser i perioder. Alle brukere

har sin egen profilside der de kan redigere generell informasjon og innstillinger. Brukere kan gi anmeldesler/tilbakemelding på hverandre sine innlegg, og vil bli varslet dersom en annen bruker skriver en anmeldelse på sin parkeringsplass. Ved registrering av parkeringsplass, bestemmer man selv pris og tiden den er tilgjengelig, og den vil automatisk bli vist for andre brukere via feeden eller profil. Man må først registrere bilen man parkerer med før man kan reservere en parkeringsplass.

## 7.2 Bedrift

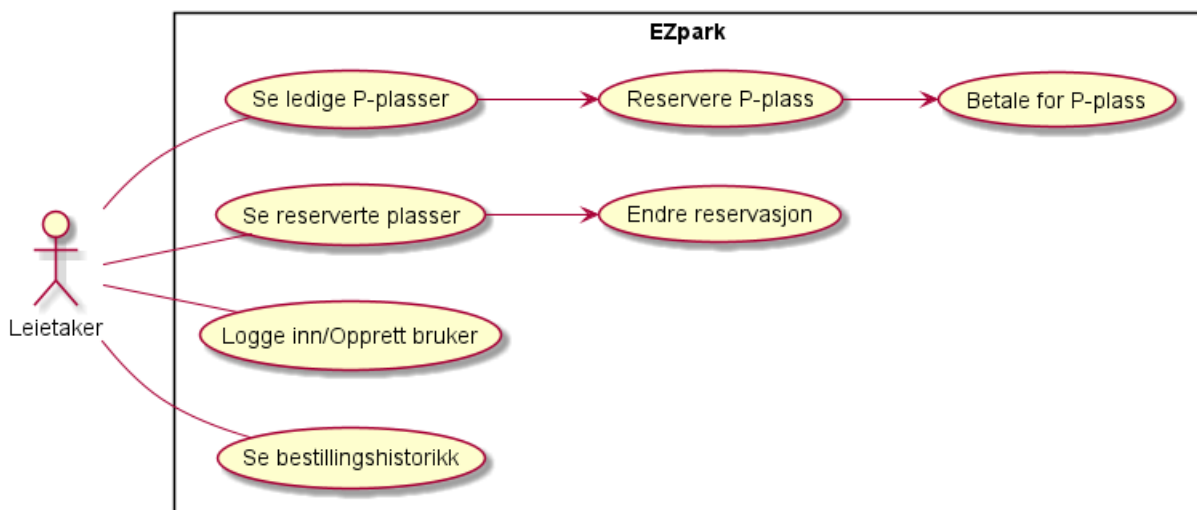
Det er kun organisasjoner som har tilgang til å opprette en bedriftskonto, da de må ha et organisasjonsnummer. Bedriftskontoen er sammensatt av flere brukerkontoer. En brukerkonto som er medlem av en bedrift har spesialtilgang til parkeringsplasser markert for generelle eller spesifikke bedrifter. Bedriftskontoen kan legge ut parkeringsplasser, men kan ikke leie en parkeringsplass. Det er kun bedrifter som kan markere parkeringsplasser for andre bedrifter.

## 7.3 Administrator

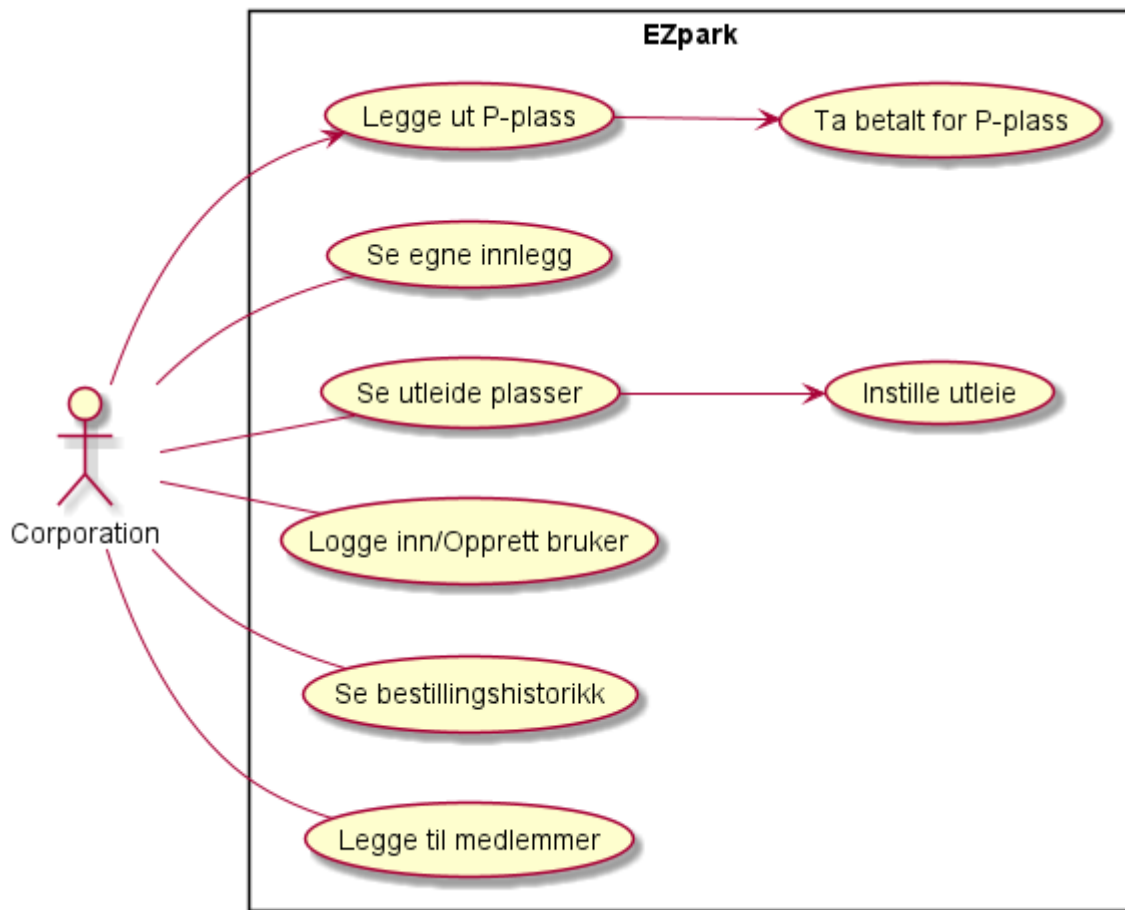
En administrator ligner en vanligbrukerkonto, men har noen ekstra rettigheter for administrering. De har full makt til å slette brukere, bedrifter, innlegg, parkeringsplasser, og anmeldelser, samt suspendere og oppheve brukerkontoer. Administrator er essensiell for å vedlikeholde en god brukeropplevelse og rettferdiggjøre systemet for alle brukere.

# 8 Use Case diagrammer

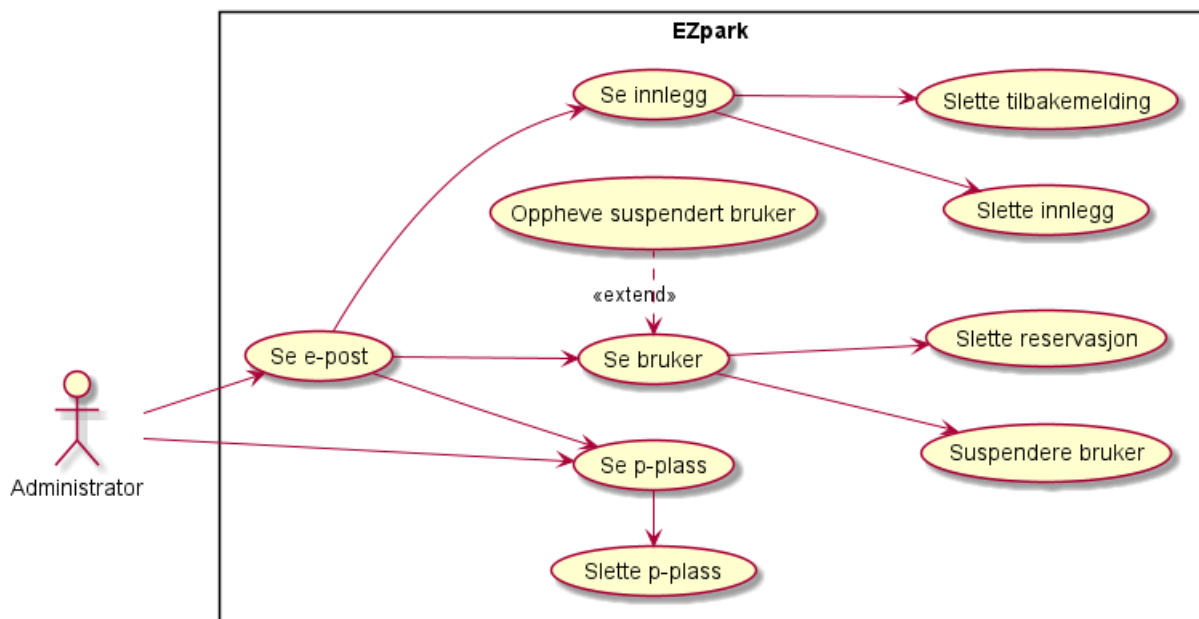
## 8.1 Use Case - Bruker (Leietaker)



## 8.2 Use Case - Bedrift



## 8.3 Use Case - Administrator





## 9 Personaer

### 9.1 Jørgen Moe



**Alder:** 34

**Jobb/Rolle:** Økonom

**Beskrivelse:**

Jobber som økonom. Han skal grave i gården og lage ny parkeringsplass. Han vet ikke hvor lang tid det kommer til å ta, og han kommer ikke til å ha et sted å parkere i mellomtiden.

**Motivasjon:**

Han har ikke parkeringsplass hjemme, så han trenger en parkeringsplass i nærheten.

**Behov:**

Han må kunne forlenge antall dager han låner en parkeringsplass

**Tekniske ferdigheter:** Gode

### 9.2 Sigurd Olson



**Alder:** 42

**Jobb/Rolle:** Ledelse i bedrift

**Beskrivelse:**

Jobber i ledelsen i en bedrift. Det er ikke alltid han får en plass til å parkere i byen når han skal på

jobb.

**Motivasjon:**

Parkeringsplassene i byen er fullstappa.

**Behov:**

Han må kunne parkere nærmere arbeidsplassen og sikre seg parkeringsplass ved et møte.

**Tekniske ferdigheter:** Gode

### 9.3 Karina Gausrud



**Alder:** 22

**Jobb/Rolle:** Uføre

**Beskrivelse:**

Karina er en rullestolbruker som trenger handicap-parkering. Hun vil helst parkere så nærme målet sitt som mulig, men vet ikke om det finnes en ledig handicap-parkeringsplass. Hun vil heller ikke risikere at stedet ikke tilbyr parkeringsplass for handicappede.

**Motivasjon:**

Hun trenger en oversikt over ledige handicap-parkeringsplasser.

**Behov:**

Hun må være sikker på at hun får en handicap-parkeringsplass

**Tekniske ferdigheter:** Gode

## 9.4 Karin



**Alder:** 34

**Jobb/Rolle:** Selger

**Beskrivelse:**

Karin er dør-til-dør selger, hun kjører til nabolagene hun jobber i hver dag, men det er vanskelig å finne parkering, som gjør at hun taper tid og penger mens hun leter. Derfor åpner hun appen og booker parkeringsplasser for områdene hun skal jobbe den neste måneden. Nå føler Karin at hun har en mindre ting å tenke på.

**Motivasjon:**

Han har ikke parkeringsplass hjemme, så han trenger en parkeringsplass i nærheten

**Mål:**

Karin ønsker å eliminere en del av hverdagen som er unødvendig og som koster henne tid og penger. Karin forventer at hun skal finne parkering, og kunne booke det for måneden fremover, eller på dagen om nødvendig.

**Reisen:**

For å oppnå dette er Karin nødt til å finne utleiere som ønsker kort-tid/dagsutleie, så hun er nødt til å søke området hun skal til, og filtrere til dette alternativet. Deretter kan hun velge sted å parkere basert på beliggenhet og pris i forhold til området hun utvalgte seg.

**Følelser:**

Karin føler seg frustrert når hun kjører rundt nabolaget hun skal banke dører i, men aldri klarer å finne et sted det er lov å parkere. Karin er på sitt lykkeligste når hun åpner appen for å sjekke hvor hun har bestilt parkering, og vet at hun ikke trenger å tenke på å lete igje

**Tekniske ferdigheter:** Gode

## 9.5 Jørn Berg



**Alder:** 56

**Jobb/Rolle:** Bonde

**Beskrivelse:**

Jørn bor i nærheten av et vakkert tjern som er populært blant unge å bade i, men siden det ikke er noen parkeringsplass der må folk parkere langs veien. Jørn eier en fin tomt i nærheten der folk kan parkere tryggere.

**Motivasjon:**

Han ønsker å tilby parkeringsplasser slik at ungdommer mer sannsynligvis kommer tilbake til tjernet og bader.

**Behov:**

Et omtalesystem som kan bidra å spre det gode rykte videre. Han vil leie ut tomten for en inntekt ved siden av gårdsdriften.

**Tekniske ferdigheter:** Dårlig

## 10 Funksjonelle krav

### 10.1 Kontoopprettelse

**Bruker skal kunne opprette en brukerkonto**

**ID: FK1**

For å opprette en brukerkonto, må bruker oppgi brukernavn, passord, e-post og telefonnummer.

**ID: FK1.1** Brukernavn og passord må være unike.

**Bedrifter skal kunne opprette en bedriftskonto**

**ID: FK2**

For å opprette en bedriftskonto, må bedriften må oppgi brukernavn, passord, e-post, bedriftsnavn og organisasjonsnummer.

**ID: FK2.1** Brukernavn, passord, bedriftsnavn og organisasjonsnummer må være unike.

**Kunden skal kunne opprette brukerkonto med administratorrettigheter**

**ID: FK3**

Bare kunde av applikasjonen kan opprette denne brukerkontoen. For å opprette denne kontoen må

kunden oppgi et brukernavn, passord og telefonnummer.

**ID: FK3.1** Brukernavn, passord og telefonnummer må være unike.

### **Systemet skal verifisere registreringen**

**ID: FK4**

Systemet må verifisere registreringen via verifiseringskode som sjekkes. Denne må bli sendt til brukeren slik at de kan skrive inn koden til programmet. Hvis koden stemmer med den som er lagret i databasen, er registreringen verifisert. Hvis ikke skal de få en feilmelding på samme siden. Etter verifisering lagres brukerdataen i databasen.

### **Dirigere opprettet bruker**

**ID: FK5**

Etter opprettelse av konto skal bruker automatisk bli ført til startsidene.

## **10.2 Innlogging**

### **Bruker skal kunne logge inn som en brukerkonto eller bedriftskonto**

**ID: FK6**

Bruker skal kunne velge å logge inn som en brukerkonto eller bedriftskonto, og så fylle ut tekst-felter med de samme opplysningene som da de opprettet kontoen sin.

### **Systemet skal verifisere innloggingen**

**ID: FK7**

Dersom innloggingsopplysningene er korrekte i forhold til opplysningene i brukeropprettingen, skal brukeren bli logget inn. Dersom de er ukorrekte, skal brukeren få en feilmelding på hva som gikk galt.

### **Dirigere innlogget brukerkonto / bedriftskonto**

**ID: FK8**

Etter vellykket innlogging av bruker/ bedrift skal brukeren bli ført til startsidene.

### **Dirigere innlogget administrator**

**ID: FK9**

Etter vellykket innlogging av administrator skal brukeren bli ført til administrator-siden.

### **Bruker skal kunne tilbakesette passord**

**ID: FK10**

Brukere har valg om å tilbakesette passord hvis de har glemt dette. I så fall skal bruker bli sendt verifiseringskode som skrives inn i et nytt felt for å verifisere at det er samme bruker. Hvis koden stemmer kan bruker skrive inn nytt passord som skal brukes fremover.

## **10.3 Verifiseringskode**

### **Systemet skal verifisere bruker gjennom en kode**

**ID: FK11**

Denne koden blir brukt til blant annet Kontoopprettelse, passordbytte og to-faktor autentisering. Hensikten er å gjenkjenne brukeren som gjør sikkerhetskritiske handlinger på sin profil. Når bruker gjør

en av disse handlingene blir koden midlertidig lagret i databasen, og videre sendt til bruker. Bruker må skrive koden inn i et tekst-felt, der den blir sjekket opp med den opprinnlige koden lagret i databasen. Hvis den stemmer er brukeren verifisert.

**ID: FK11.1**    Koden skal være fire-sifret

**ID: FK11.2**    Koden skal bli sendt via sms

## **10.4    Presentasjon av parkeringsplasser (Feed)**

**Feeden skal oppliste relevante parkeringsplasser**

**ID: FK12**

Alle brukertyper skal ha tilgang til en opplistingen som består av innlegg fra andre brukere. Det er her brukere ettersøker og reserverer parkeringsplasser. Standaropplistingen rangeres etter nærmest avstand fra posisjonen til brukeren.

**ID: FK12.1**    Posisjonen hentes via cookies.

**Bruker skal kunne sette et filter på feeden**

**ID: FK13**

Opplistingen rangeres via et brukervalgt filter. Filteret bestemmer om opplastingen rangeres basert på nærmest avstand, lavest til høyest pris, høyest til lavest pris, parkeringsplasstype og om den leies ut av en brukerkonto eller bedriftskonto.

**Bruker skal kunne søke etter en lokasjon**

**ID: FK14**

Feeden skal rangere parkeringsplassene basert på avstand i forhold til denne lokasjonen.

**ID: FK14.1**    Søkefeltet skal akseptere adresse.

**ID: FK14.2**    Søkefeltet skal akseptere koordinater.

## **10.5    Leie parkeringsplass**

**Bruker skal kunne leie parkeringsplass som en brukerkonto**

**ID: FK15**

Brukerkonto skal kunne leie en eller flere parkeringsplasser. Dette gjøres ved at systemet oppretter en reservasjon for der parkeringsplassen knyttes opp til brukeren som leier den.

**Applikasjonen skal varsle utløpstid**

**ID: FK16**

Når en reservert parkeringsplass holder på å gå tom for tid, så skal brukeren som har leid parkeringsplassen få en varsel om at den holder på å gå ut minst 30 minutter før.

**ID: FK16.1**    Varselet skal gå gjennom SMS og push-notifikasjoner

**Bruker skal ha tilgang til en oversikt over reserverte parkeringsplasser**

**ID: FK17**

Denne oversikten viser alle parkeringsplasser som denne brukeren har reservert sortert fra nyest til eldst.

## 10.6 Tilbakemelding på innlegg (Anmeldelser)

### Bruker skal kunne gi tilbakemelding på et innlegg

#### ID: FK18

Dersom brukeren har brukt en parkeringsplass, skal de kunne gi tilbakemelding på innlegget til den parkeringsplassen. Innlegget må minst bestå av en poengsum. Om ønskelig kan bruker også legge til en kommentar. Tilbakemeldingen skal lagres i databasen og tilknyttes innlegget.

**ID: FK18.1** Hver bruker kan bare skrive en anmeldelse per innlegg.

**ID: FK18.2** Kommentaren skal være på maks 130 tegn.

**ID: FK18.3** Poengsummen skal være fra 1 til 5, der 5 er den høyeste poengsummen.

### Tilbakemeldinger skal vises på hvert innlegg

#### ID: FK19

Alle anmeldelser som tilhører et innlegg skal vises på listeform på innlegget, der den nyeste tilbakemelding ligger øverst.

**ID: FK19.1** Hver tilbakemelding skal vise poengsummen, kommentaren og brukernavn til de som lagde den.

### Utleier skal kunne få tilbakemeldingene via e-post

#### ID: FK20

Ved valg har utleier mulighet til å få tilsendt alle tilbakemeldinger på e-post.

## 10.7 Opprette parkeringsplasser

### Brukerkonto og bedriftskonto skal kunne registrere en eller flere parkeringsplasser

#### ID: FK21

For å registrere en parkeringsplass må følgende oppgis:

- Bredde, lengde, høyde i centimeter
- Antall parkeringsplasser
- Type parkeringsplass (om det er handikapparkering, elektriskparkering, osv.)
- Etasje
- Pris (dette skal kunne endres manuelt eller automatisk i etterkant)
- Postadresse
- Gateadresse
- Gatenummer

Etter opprettelsen lagres parkeringsplassen i databasen.

**ID: FK21.1** Bedriftskonto skal kunne markere en parkeringsplass for en spesifikk bedrift eller for bedrifter generelt

**ID: FK21.2** Parkeringsplassen skal knyttes til en unik ID

**ID: FK21.3** Parkeringsplassen skal knyttes til brukeren som opprettet den.

**Bruker skal ha tilgang til en oversikt over sine opprettede parkeringsplasser**

**ID: FK22**

Denne oversikten viser alle opprettede parkeringsplasser fra denne brukeren sortert fra nyest til eldst.

## **10.8 Leiebetingelser**

**Pris per time skal bestemmes av utleier**

**ID: FK23**

Dette er prisen som parkeringsplassen leies ut for (per timer) og bestemmes av bruker som registrerte parkeringsplassen. Den bestemmes på hver individuelle parkeringsplass.

**ID: FK23.1** Prisen skal kunne oppdateres etter opprettelsen.

**Tiden skal bestemmes av bruker**

**ID: FK24**

Dette er fra-tid og til-tid som parkeringsplassen kan leies av andre brukere.

**Brukere skal kunne se ut ifra innlegget om parkeringsplassen er ledig eller ikke**

**ID: FK25**

Ledig-statusen regnes ut dersom den tilgjengelige tiden samsvarer med nåværende klokkeslett.

**ID: FK25.1** Hvis en annen bruker låner parkeringsplassen, så skal tiden den er opptatt bli trukket fra det totale tilgjengelige tidsintervallet.

## **10.9 Betaling**

**Bruker skal betale utleieren for å opprette en reservasjon**

**ID: FK26**

Bruker betaler for antall timer de ønsker å leie parkeringsplassen.

## **10.10 Administrering**

**Administrator skal kunne bannlyse bruker**

**ID: FK27**

Administrator skal ha full rett til å bannlyse en vilkårlig bruker.

**ID: FK27.1** Den bannlyste brukeren skal få beskjed om bannlysingen via e-post og direkte i systemet.

**Administrator skal kunne slette bruker**

**ID: FK28**

Administrator skal ha full rett til å slette individuelle brukere slik at all tilhørende data blir også slettet fra databasen.

**ID: FK28.1** Den slettede brukeren skal få beskjed om slettingen via e-post.



## **Administrator skal kunne slette tilbakemeldinger**

### **ID: FK29**

Administrator skal ha full rett til å slette individuelle tilbakemeldinger på andre innlegg slik at de fjernes både fra grensesnittet og databasen.

## **Administrator skal kunne slette innlegg**

### **ID: FK30**

Administrator skal ha full rett til å slette individuelle innlegg.

**ID: FK30.1** Bruker som lagde innlegget skal få beskjed om slettingen via e-post og direkte i systemet.

## **Administrator skal kunne slette parkeringsplasser**

### **ID: FK31**

Administrator skal ha full rett til å slette individuelle parkeringsplasser.

**ID: FK31.1** Bruker som opprettet parkeringsplassen skal få beskjed om slettingen via e-post og direkte i systemet.

## **Administrator skal kunne slette reservasjoner**

### **ID: FK32**

Administrator skal ha full rett til å slette individuelle reservasjoner.

**ID: FK32.1** Bruker som opprettet reservasjoner skal få beskjed om slettingen via e-post og direkte i systemet.

## **Bedrift skal ha administrerende rettigheter på deres egne innlegg**

### **ID: FK33**

Dette innebærer at bedriften kan blokkere brukere av deres egne parkeringsplasser fra å bruke de igjen og slette tilbakemeldinger på deres egne innlegg.

## **Bruker skal kunne rapportere andre brukere sine innlegg**

### **ID: FK34**

Brukere har frihet til å rapportere andre brukere sine innlegg. Rapporteringen skal bestå av en begrunnelse.

**ID: FK34.1** Brukeren som har lagd innlegget skal få varsel via e-post om rapporteringen.

## **Bruker skal kunne rapportere andre brukere**

### **ID: FK35**

Brukere har frihet til å rapportere andre brukere. Rapporteringen skal bestå av en begrunnelse.

**ID: FK35.1** Brukeren som har lagd innlegget skal få varsel via e-post om rapporteringen.

## **Bruker skal kunne slette sine reservasjoner**

### **ID: FK36**

Bruker har frihet til å slette en reservasjon de har opprettet.

**ID: FK36.1** Utleieren av parkeringsplassen få en varsel på at parkeringsplassen ikke lenger er reservert.

**ID: FK36.2** Innlegget skal oppdatere seg slik at reservasjonen fjernes, og er nå ledig.

## **Bruker skal kunne slette sine parkeringsplasser**

### **ID: FK37**

Bruker har frihet til å slette parkeringsplasser de har opprettet.

**ID: FK37.1** Innlegget for denne parkeringsplassen skal også slettes.

## **Bruker skal kunne slette sine innlegg**

### **ID: FK38**

Bruker har frihet til å slette sine innlegg

**ID: FK38.1** Feed skal oppdateres slik at dette innlegget ikke blir opplistet.

## **10.11 Persistent data**

### **Systemet skal lagre informasjon om parkeringsplasser**

#### **ID: FK39**

Systemet skal lagre all informasjon relatert til enhver parkeringsplass i en tabell i en database, slik at endringene gjenstår.

### **Systemet skal lagre informasjon om innlegg**

#### **ID: FK40**

Systemet skal lagre all informasjon relatert til hvert innlegg i en tabell i en database, slik at endringene gjenstår.

### **Systemet skal lagre informasjon om brukere**

#### **ID: FK41**

Systemet skal lagre nødvendig innlogging og brukerinformasjon for at bruker skal kunne ha bruk av systemet. Dette skal lagres i en tabell i en database, slik at endringene gjenstår.

### **Systemet skal lagre informasjon om reservasjoner**

#### **ID: FK42**

Systemet skal lagre all informasjon relatert til enhver reservasjon i en tabell i en database, slik at endringene gjenstår.

## **11 Ikke-funksjonelle krav**

### **11.1 Tilgjengelighet**

#### **Bruker skal ha tilgang til applikasjonen**

##### **ID: IFK1**

All brukere skal ha tilgang til applikasjonen til enhver tid, overalt i landet.

#### **Bruker må kvalifiseres ved tilgang til applikasjonen**

##### **ID: IFK2**

Brukere ska være minimum 16 år og ha gyldig førerbevis for å få tilgang til applikasjonen.

## **11.2 Sikkerhetskopi**

**Kunde skal ta delvis backup av database**

**ID: IFK3**

Delvis sikkerhetskopiering av database skal skje hver 24 time kl 04.00.

**Kunde skal ta full backup av database**

**ID: IFK4**

Full sikkerhetskopiering av database skal skje minst en gang i måneden.

## **11.3 Tredjepart**

**Betaling skal gå gjennom en tredjepart**

**ID: IFK5**

Betaling verifiseres og håndteres av NETS

## **11.4 Brukerdata og sikkerhet**

**Applikasjonen skal opplyse om vilkår for bruk**

**ID: IFK6**

Brukeren skal få en oversikt over informasjon knyttet til bruksområder for deres brukerdata, hvilke rettigheter de har, og hvilke rettigheter kunden har ovenfor deres bruk av tjenesten. Denne oversikten skal være på form av TOS (Terms of Service).

**Applikasjonen skal utdype om personvern**

**ID: IFK7**

Lagring av persondata skal skje i henhold med oppdatert GDPR-lovverk.

**Applikasjonen skal følge en standard for sikkerhet**

**ID: IFK8**

Programvareutviklingen skal skje i samråd med datatilsynets anbefaling om innebygd personvern.

**ID: IFK8.1** Applikasjonen skal minst følge standarden NS-EN ISO/IEC 27000.

## **11.5 Programvare**

**Tidsrom ved oppstart skal være tilstrekkelig**

**ID: IFK9**

Oppstart av Javalin til localhost:7000 skal ta mellom 800 - 900 ms.

**Tidsrom ved sidebytte skal være tilstrekkelig**

**ID: IFK10**

For at å bevare brukeropplevelse må tiden det tar for å utføre handlinger på applikasjonen være begrenset. Det innebærer at applikasjonen ikke henger seg opp.

**ID: IFK10.1** Lasting av side skal ta minimum 2 sekund.

**Programvaren skal kunne støtte drastiske endringer**

**ID: IFK11**

Systemet skal være lag-delt slik at det er kost-effektivt å vedlikeholde eller gjøre endringer.

**Viktige deler av programvaren skal testes**

**ID: IFK12**

Alle metoder/funksjoner som er ofte i bruk skal ha tester for å sikre at kjernelogikken stemmer.

## **11.6 Samsvar**

**Parkeringsplasser må samsvare med retningslinjer**

**ID: IFK13**

Parkeringsplasser som er registrert må følge nasjonale, regionale og lokale retningslinjer for manøvreringsrom og areale for parkeringsplass.

## **11.7 Avhengigheter**

**Systemet skal bruke JavaJDK 11.**

**ID: IFK14**

Systemet skal bruke JavaJDK 11 for å kompilere og kjøre back-end koden.

**Systemet skal bruke Maven**

**ID: IFK15**

Systemet skal bruke Maven 3.6.3 som byggeautomatiseringsverktøy for å gjøre det enklere å hente inn avhengigheter.

**Systemet skal bruke Javalin**

**ID: IFK16**

Systemet skal bruke Javalin 3.11.0 som Web rammeverk for lokal webserver

**Systemet skal bruke Java JSON biblioteket Jackson**

**ID: IFK17**

Systemet skal bruke Jackson 2.11.3 Java JSON library for serialisering og deserialisering av data i prototypen

**Systemet skal bruke JUnit-Jupiter**

**ID: IFK18**

Systemet skal bruke JUnit-Jupiter 5.7.0 Rammeverk for testing i Java

**Systemet skal bruke Vue.js**

**ID: IFK19**

Systemet skal bruke Vue.js 2.6.10 som Open-source Front-end modell-visning-modellvisningsrammeverk for å bygge brukergrensesnitt.

## Systemet skal ha testing for web

### ID: IFK20

**ID: IFK20.1** Systemet skal bruke Selenium 3.141.59 Rammeverk for testing av web applikasjoner.

**ID: IFK20.2** Systemet skal bruke Webdrivermanager 4.2.2 Bibliotek for installasjon av web-drivere for bruk i testing.

**ID: IFK20.3** Systemet skal bruke Unirest 1.4.9 Bibliotek for HTTP klienter brukt i testing.

## 12 Prototype

### 12.1 Sammendrag

Vi har laget en forenklet versjon som viser hvordan strukturen på programmet kan se ut. Vårt fokus ved prototypen har vært å vise en oversikt over parkeringsplasser der man kan trykke inn på en spesi-fikk parkeringsplass og få mer informasjon om selve parkeringsplassen, og reservere den.

Det andre fokuset var å ha en admin bruker som kan kontrollerer bruken av siden ved å suspendere og slette brukere, parkeringsplasser og reserverasjoner.

I denne forenklete versjonen, gjør vi all lagring i json filer.

Vi har delt back-end koden i forskjellige deler som kalles controller, model, datahandler og repository. Dette gjør at funksjonaliteten splittes til forskjellig seksjoner av programmet og tillater endringer uten å påvirke den generelle funksjonaliteten i programmet. Dette gjør for eksempel at et ombytte til en database-modell ville kun trenge endring i datahandler seksjonen av programmet.

Vi har valgt å bruke Javalin med Vue som rammeverk til front-end. Dette har gjort det enkelt å gjøre MVP-en pen og oversiktlig, og er den største avhengigheten i programmet vårt.

### 12.2 UML

Intensjonen med disse UML-ene var å ha en oversikt over hvordan koblingene mellom klassene skulle fungere. Selvfølgelig har prototypen endret seg i løpet av utviklingsprosessen, men de opprinnlige UML-diagrammene var et fint startpunkt.

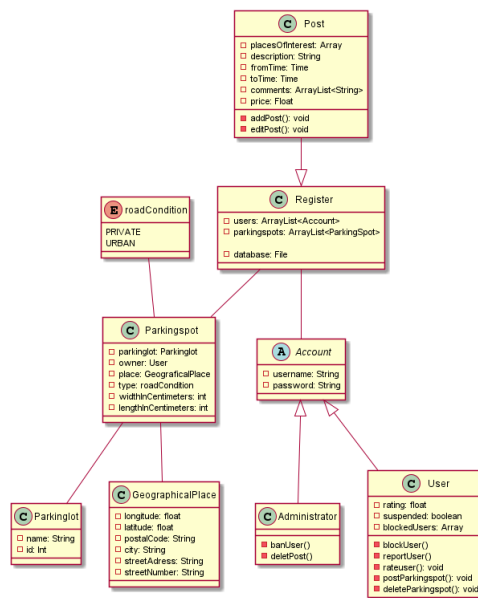


Figure 1: Dette diagrammet viser den originale planlagte klassestrukturen for MVP-en

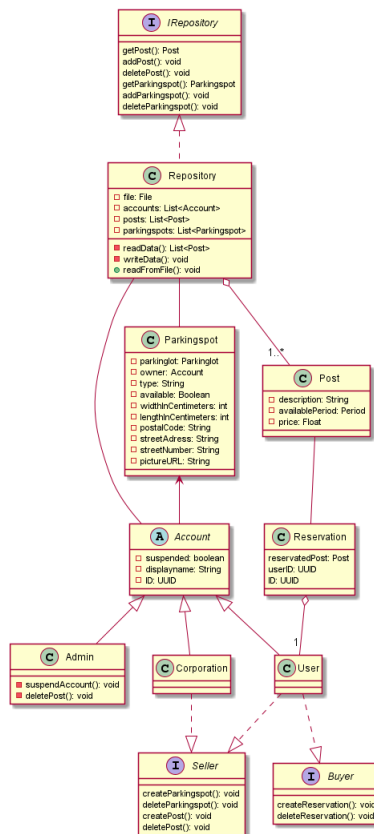


Figure 2: Dette diagrammet viser klasse strukturen og koblingen i MVP-en

## 12.3 Innstallasjon

For å bygge og kjøre programmet, høyre-klikk mappen og velg å åpne som et intelliJ prosjekt, eller åpne intelliJ og trykk på fil, velg åpne mappe prosjekt, velg mappen og last inn prosjektet. Da vil også alle avhengigheter bli installert, eller det vil komme en installasjonsprompt med beskrivelse i terminalen i intelliJ. Dersom dette skjer, må man fullføre installasjonen som står beskrevet i terminalen. Deretter trykker man på den grønne hammeren i høyere hjørnet eller trykker ctrl + F9 for å bygge programmet. Om det fortsatt ikke kjører, last inn maven avhengigheter på nytt ved å trykke på søkeknappen i høyre hjørne av intelliJ, skriv maven, og trykk reload maven dependencies.

## 12.4 Bruksmanual

For å starte programmet velg main klassen i src mappen, velg så main metoden og trykk grønn pil ved siden av Public Class Main, for å kjøre run configuration trykk på add configuration, trykk så på application og velg run. Trykk shift + F10, eller grønn pil ved siden av run configuration. Da vil javalin kjøre i konsollen i intelliJ, og det åpnes en port til <http://localhost:7000/>.

Når man har kommet inn til localhost:7000 blir man presentert med tre knapper: En for bruker, en for corporation og en for admin. Bruker-knappen leder til oversikten av parkeringsplasser der alle tilgjengelige plasser vises. Derfra kan bruker trykke seg inn på en spesifikk parkeringsplass, og videre trykke seg inn til bestillings-siden, fylle ut skjemaene, og trykke submit knappen for å bestille. Da skal parkeringsplassen dukke opp i “My parking spots” på bunnen av siden der det står reservasjoner.

Inne på denne siden kan brukeren opprette parkeringsplasser de ønsker å leie ut ved å trykke på “Create new parking spot”-knappen der de tas til et utfyllingsskjema. Dersom skjemaet ikke er tilstrekkelig utfyllt vil en feilmelding vises med all info som er nødvendig i skjemaet. Når skjema er riktig utfyllt kan man trykke submit knappen, og den nye parkeringsplassen vil umiddelbart bli lagt ut til både hovedoversikten og til “My parking spots”-siden. Foreløpig i prototypen, fungerer corporation-brukeren på lik linje som bruker-typen.

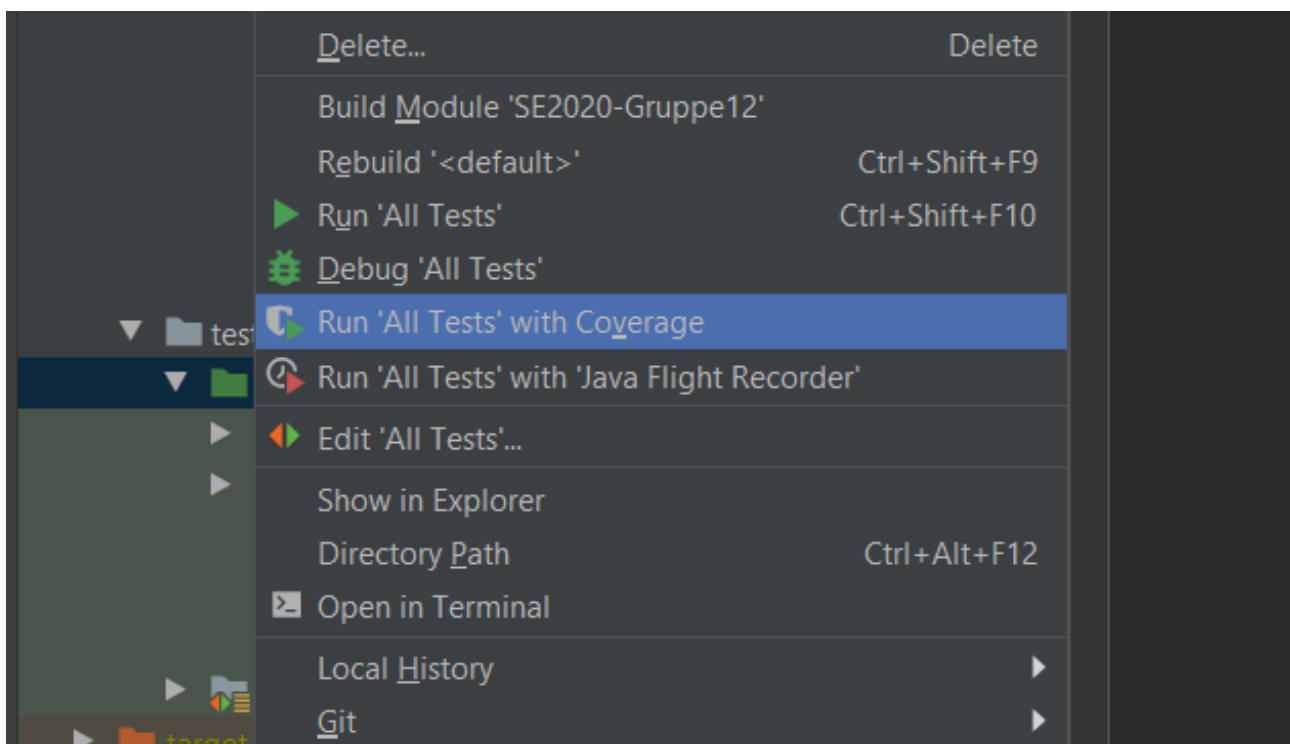
Dersom man trykker admin-typen kommer man til en oversikt over brukere, parkeringsplasser, poster og reservasjoner. Admin kan suspendere eller slette kontoer, og i tillegg slette parkeringsplasser, poster og reservasjoner.

## 12.5 Kjøre tester

For å kjøre testene må man åpne test-mappen i IntelliJ og høyre-klikke på den grønne java-mappen og velg Run ‘All Tests’ With Coverage. Da kjører testene og man kan se andelen av programmet der det er test-dekning.

Man bør ikke gjøre endringer inne i applikasjonen da dette kan medføre feil i kjøring av repository tester.

Det går ikke å ha to instanser av applikasjonen kjørende samtidig da den vil prøve å opprette en port som allerede eksisterer og tester vil ikke bli kjørt.



## 12.6 Implementerte krav med tester

Krav for prototypen går mye på grunnleggende funksjonalitet. Krav som ikke er essensielle for å tilfredstille kunden har blitt mindre prioritert her. Oppstillingen nedenfor består først av ID-en, kravet og testene i prototypen som tester kravet.

Her er en de implementerte kravene:

- **FK6** - Bruker skal kunne logge inn som en brukerkonto eller bedriftskonto

*Tester*

- doesPageLoad, frontPageLoads()

- **FK8** - Dirigere innlogget brukerkonto / bedriftskonto

*Tester*

- doesPageLoad, parkingSpotOverviewPageLoads()
- isRedirectedToCorrectURL, isUserRedirectedToFrontPage()

- **FK12** - Presentasjon av parkeringsplasser (Feed)

- **FK15** - Bruker skal kunne leie parkeringsplass som en brukerkonto

*Tester*

- doesPageLoad, parkingSpotDetailPageLoads()

- **FK21** - Brukerkonto og bedriftskonto skal kunne registrere en eller flere parkeringsplasser

Denne er implementert, med unntak på variabler for etasje, pris og antall.

*Tester*

- doesPageLoad, createParkingSpotPageLoads()
- userEndToEnd, userCreatesParkingSPot()

- **FK26** - Bruker skal betale utleieren for å opprette en reservasjon

*Tester*

- doesPageLoad, parkingSpotPaymentPageLoads()
- userEndToEnd, userCreatesReservation()

- **FK17** - Bruker skal ha tilgang til en oversikt over reserverte parkeringsplasser

*Tester*

- doesPageLoad, myParkingSpotPageLoads()

- **FK22** - Bruker skal ha tilgang til en oversikt over sine opprettede parkeringsplasser

*Tester*

- doesPageLoad, myParkingSPotPageLoads()

- **FK36** - Bruker skal kunne slette sine reservasjoner

*Tester*

- userEndToEnd, userDeletesReservation()

- **FK37** - Bruker skal kunne slette sine parkeringsplasser

*Tester*



- userEndToEnd, userDeletesParkingSpot()
- **FK27** - Administrator skal kunne bannlyse bruker
- **FK28** - Administrator skal kunne slette bruker
- **FK30** - Administrator skal kunne slette innlegg
- **FK31** - Administrator skal kunne slette parkeringsplasser
- **FK32** - Administrator skal kunne slette reservasjoner
- **FK9** - Dirigere innlogget administrator

*Tester*

- isRedirectedToCorrectURL, isAdminRedirectedToAdminPage()
- doesPageLoad, adminPageLoads()

## 12.7 Avhengigheter

Avhengigheter blir automatisk lastet inn av maven ved åpning av programmet. Dersom dette ikke skjer kan man gjøre det manuelt i IntelliJ.

## 12.8 Begrensninger

De begrensningene vi har møtt på i utviklingen har for det meste vært praktiske. Vi har valgt å holde corporation og bruker som like bruker-typer i prototypen ettersom at de i hovedsak deler mange egenskaper.

Vi møtte motstand ved lastning av flere forskjellige typer data til samme vue side, som har limitert dens totale omfang for visning, og reservasjon. Dette spesifikt da single-parking-spot seksjonen av programmet.

Vi har forsøkt å begrense prototypen til skallet som er nødvendig før implementering av database, betalingssystem og brukerregistreringer.

Vi har ikke lagt inn bil-registrering eller brukerregistrering da dette ikke er nødvendig for funksjonaliteten i prototypen.

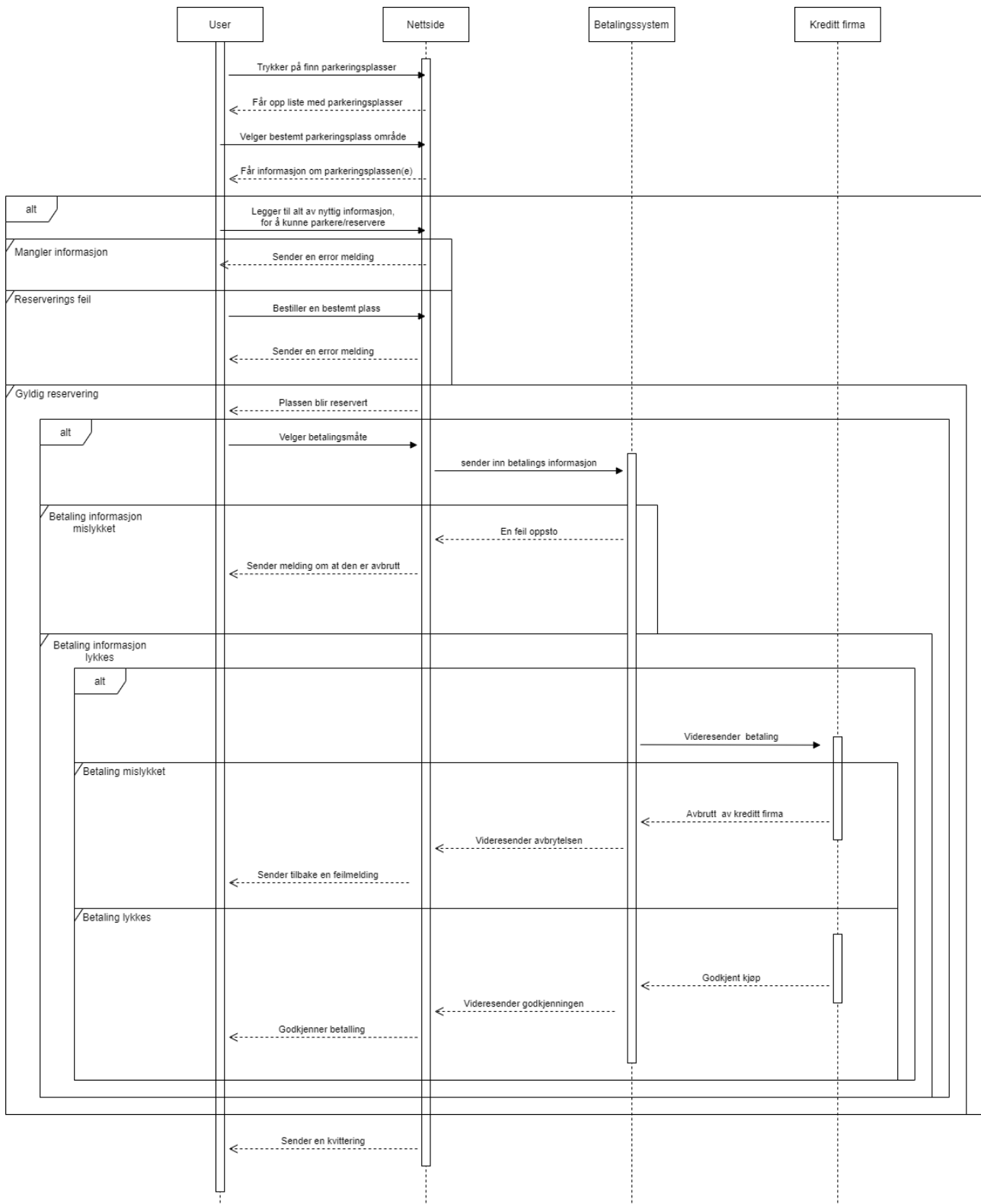
Vi ønsket å legge inn corporation-bruker som kun kunne gjøre utleieing av p-plass, men også som kunne legge til medlemmer som kan leie av de plassene spesifikt. Dette ble en for stor utfordring, og vi valgte å holde den i sin basis-form.

Det var ment at vi skulle ha flere brukere til prototypen, men valgte å fokusere på å få én bruker til å fungere på alle stegene i utlegging og leie-prosess.

MVP-en har ikke fått inn noe rangeringssystem eller filtrering, da vi ønsker å filtrere på lokasjon, men ikke har kunnskapen til å gjøre denne typen filtrering.

## 13 Diagrammer

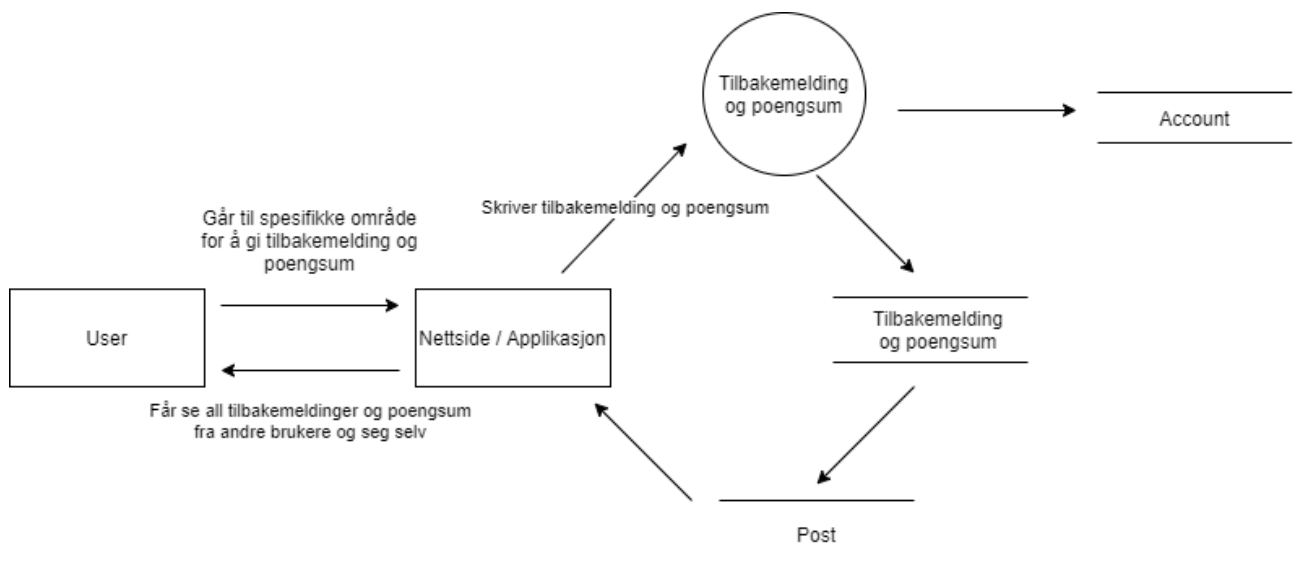
### 13.1 Sekvensdiagram - Leie en parkeringsplass.



#### Forklaring

Når en bruker har kommet seg inn på applikasjonen så skal de kunne trykke på en knapp som fører dem til et sted som sørger for at de skal søke/lete etter en parkeringsplass. Etter de har funnet en parkeringsplass så skal de fylle inn enkelt informasjon om parkeringen for å bestille den. Hvis bestillingen går igjennom så skal de få tilbake en kvittering/bevis på at alt har fungert og at de leier den parkeringsplassen de har betalt for.

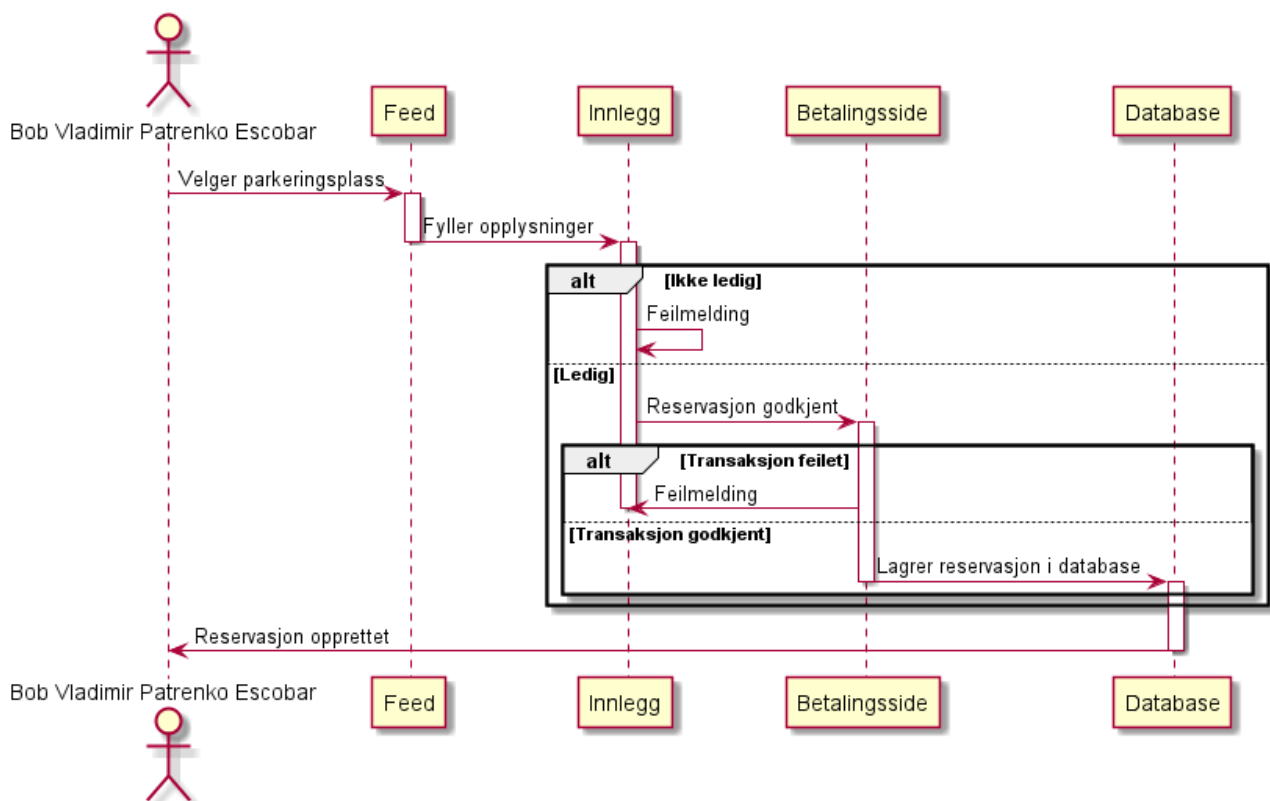
## 13.2 Dataflytdiagram - Tilbakemelding



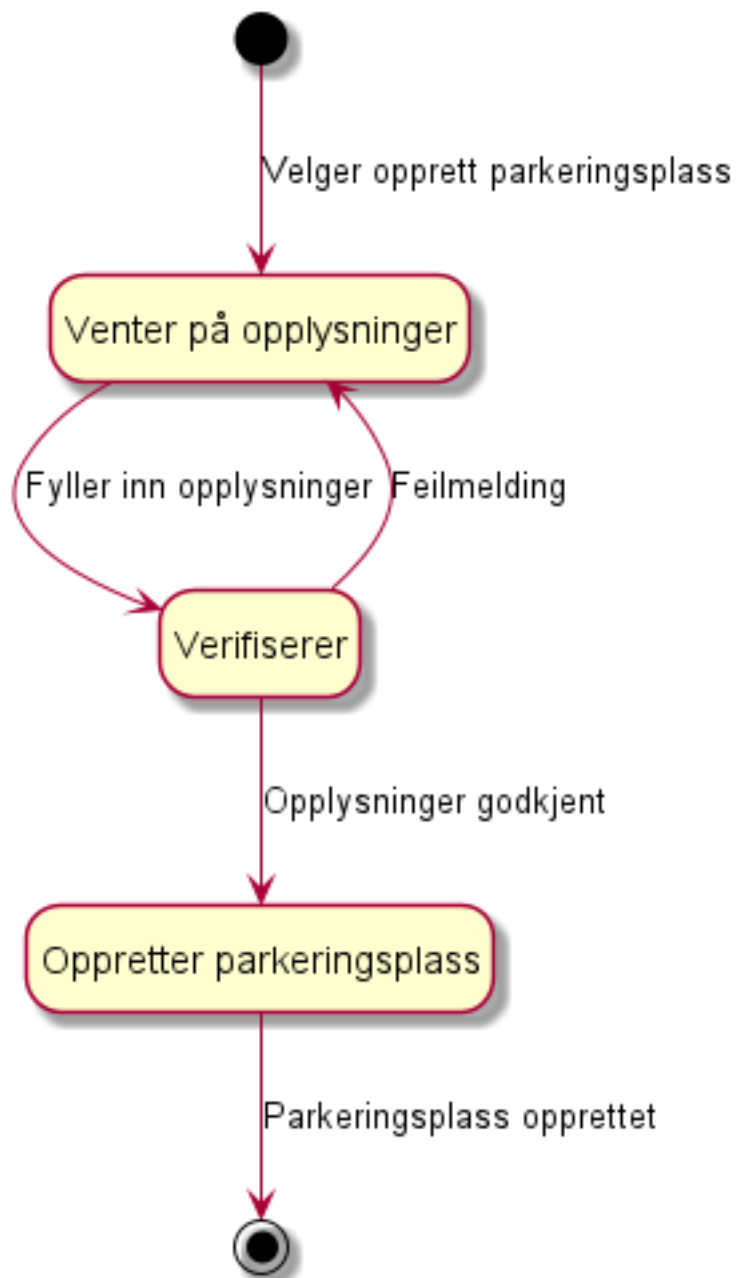
### Forklaring

Etter at en bruker har leie parkeringsplass hos en utleier så skal de kunne gi tilbakemelding og en poengsum, gjennom en applikasjonen. Når de gir en tilbakemelding så blir den lagret i en database (Tilbakemelding og poengsum) som er knyttet til den parkeringsplassen (post, databasen) og tilbakemelding skal være synlig for brukeren selv og andre, man kan også se hvem som har skrevet hva.

## 13.3 Sekvensdiagram - Opprette reservasjon



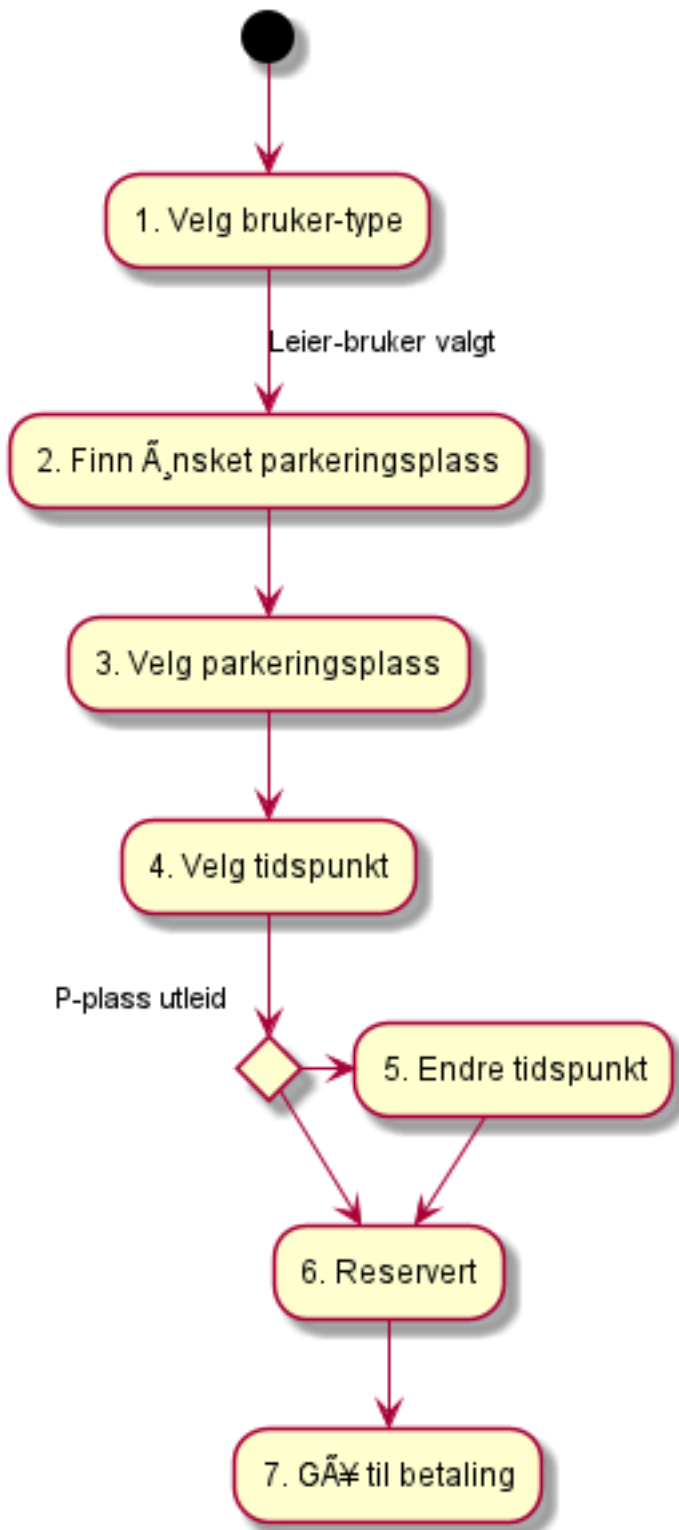
### 13.4 Tilstandsdiagram - Registrere parkeringsplass



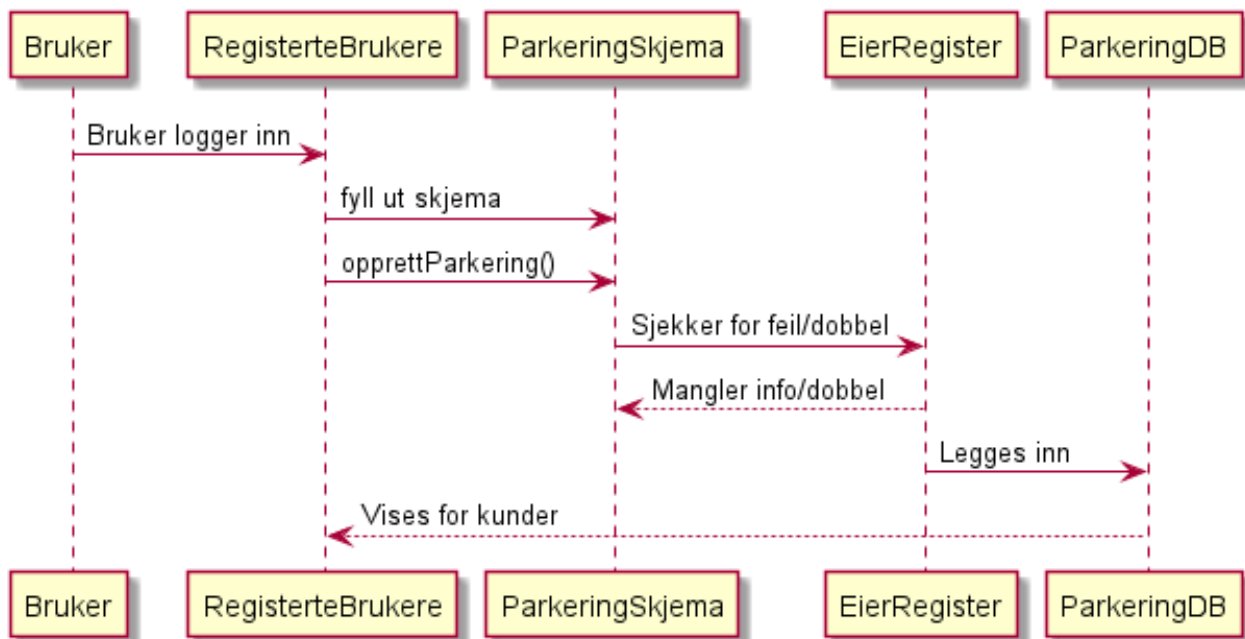
#### Forklaring

Når bruker oppretter parkeringsplass venter systemet på en POST forespørsel med informasjon om gatenavn, adresse, bredde, lengde og høyde, samt informasjon angående parkeringsplasstype (handicap, el-bil, osv.). Dersom opplysningene ikke tilfredstiller forventningene i systemet, får bruker en feilmelding og blir nødt til å fylle ut på nytt. Dersom opplysningene er godkjent blir parkeringsplassen opprettet.

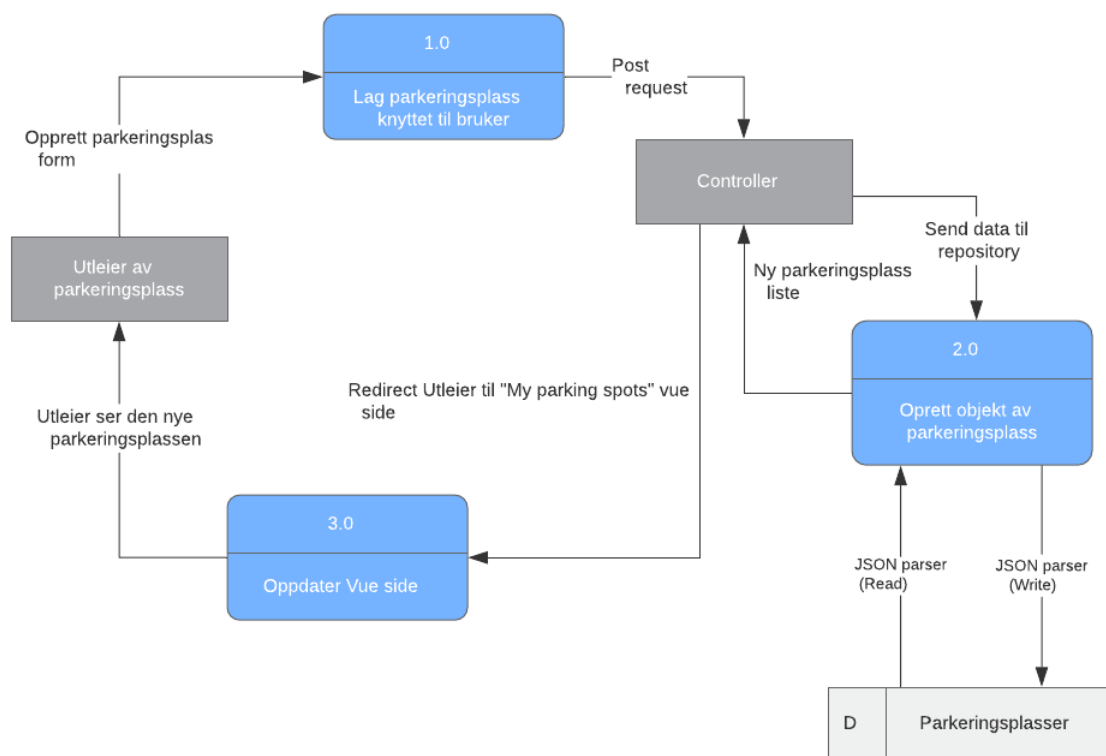
### 13.5 Aktivitetsdiagram - Leie en parkeringsplass



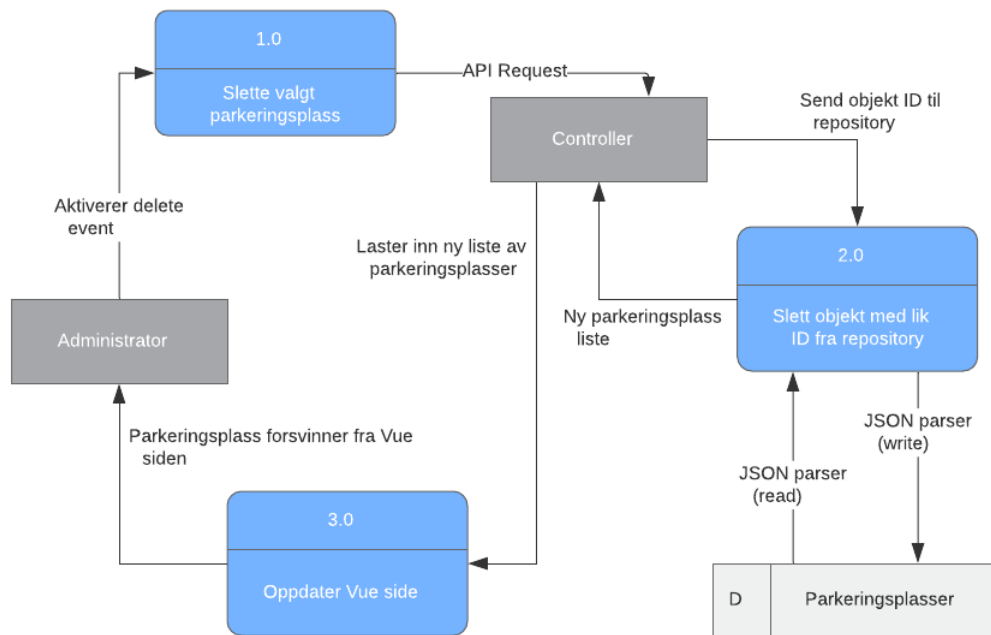
## 13.6 Sekvensdiagram - Leie ut en parkeringsplass



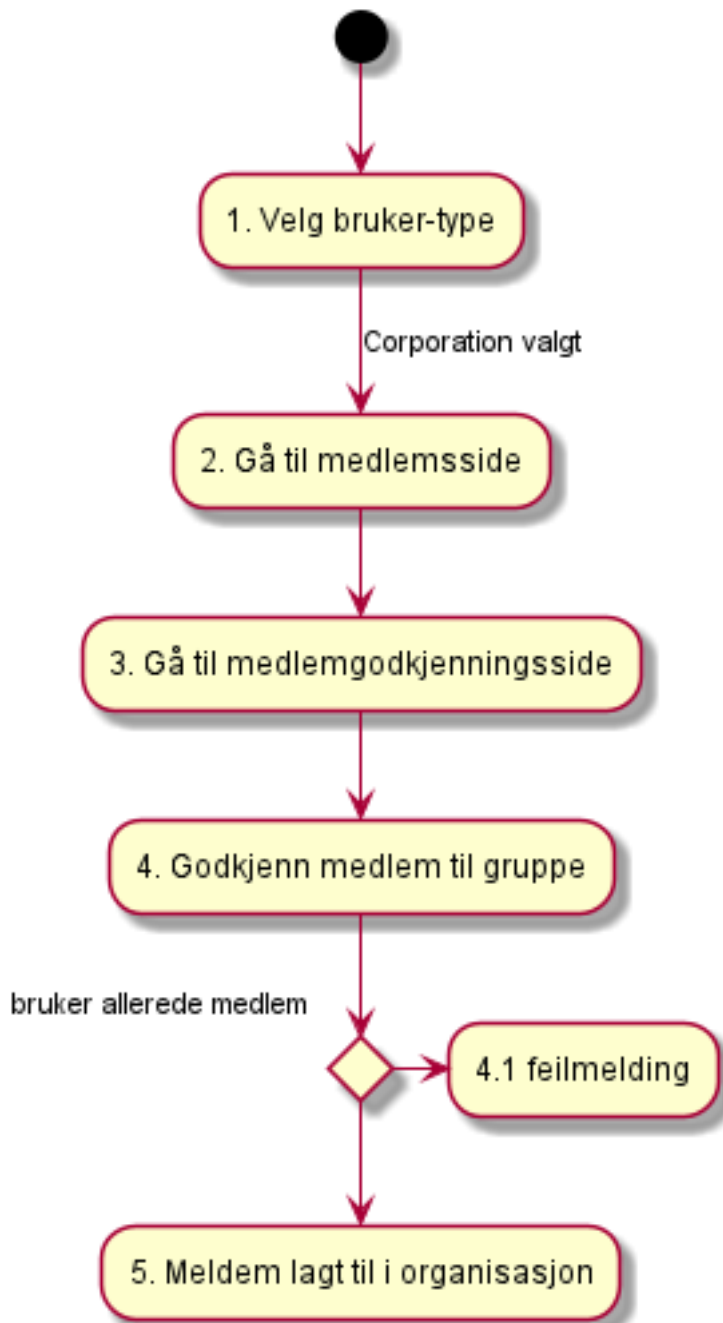
## 13.7 Dataflytdiagram - Opprette parkeringsplass



## 13.8 Dataflyttdiagram - Slette Parkeringsplass

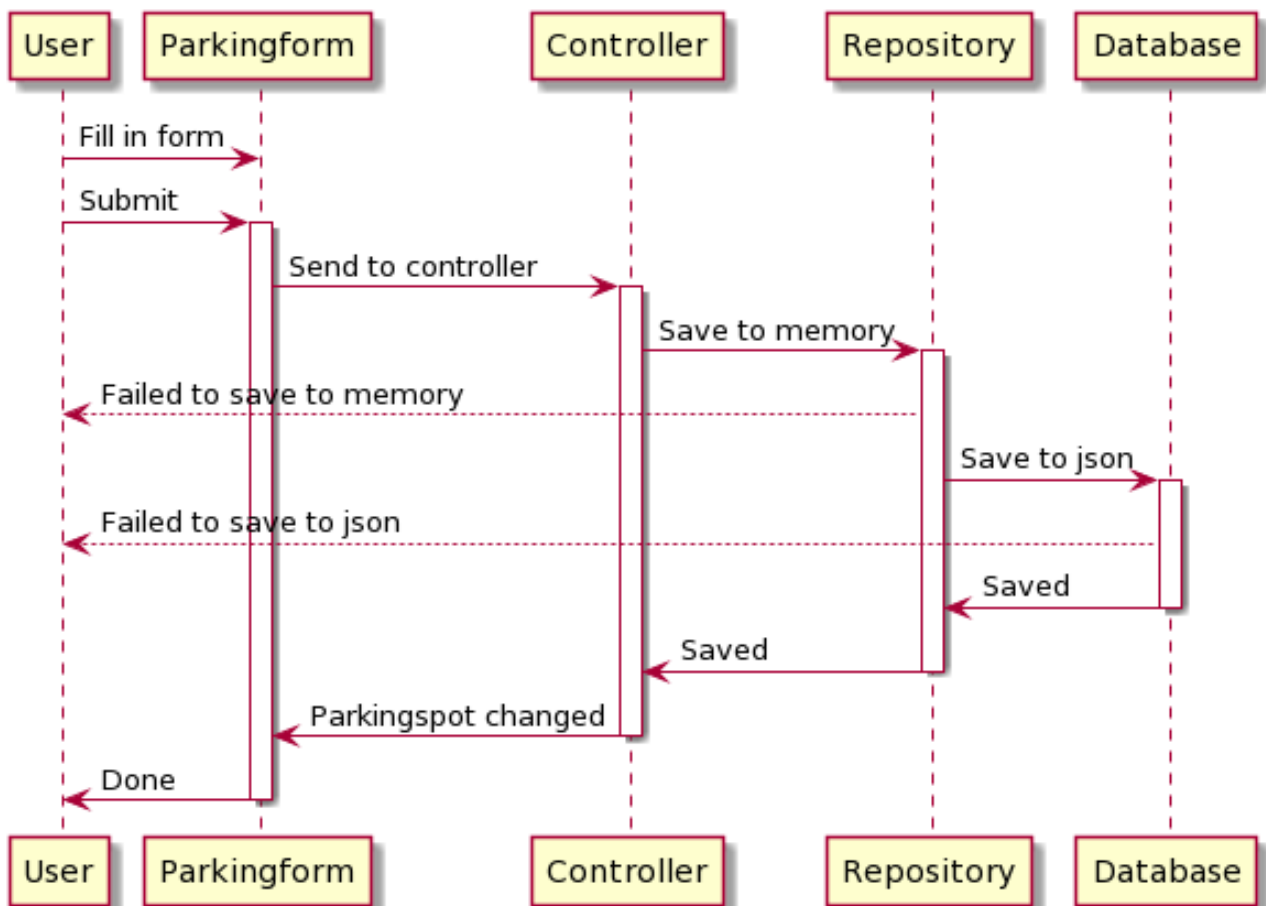


### 13.9 Aktivitetsdiagram - Legge til medlem i organisasjon





## 13.10 Sekvensdiagram - Oppdatere Parkeringsplass



## 14 Estimering

Denne seksjonen blir brukt til å visualisere estimeringen for hvert funksjonelt og ikke-funksjonelt krav. Verdiene for viktighet og utviklingstid kommer i utgangspunktet fra erfaringen på prototypen, særdeles de kravene som faktisk er implementert. Det er også blitt tatt noe inspirasjon fra eksisterende tjenester som Easypark for å bedømme vanskelighetsgraden på enkelte krav, hovedsakelig ikke-funksjonelle krav.

### 14.1 Størrelser

Disse størrelsene blir brukt til å visualisere prioriterten på hvert krav. I prinsippet er krav med 0 på størrelse nøytrale, det vil si at dersom den er like viktig som tiden det tar å utvikle den (i henhold til t-shirt sizing), så er den nøytral. Krav med størrelse over 0 er mer prioriterte enn krav under 0. Når det er sagt er det fremdeles verdt å inspisere både viktighet og utviklingstid på hvert krav.

	Utviklingsstørrelse			
Verdi	X-Large	Large	Medium	Small
X-Large	0	4	6	7
Large	-4	0	2	3
Medium	-6	-2	0	1
Small	-7	-3	-1	0

## 14.2 Estimeringstabell - Funksjonelle krav

### Kontoopprettelse

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK1	Bruker skal kunne opprette en brukerkonto	M	M	0
FK2	Bedrifter skal kunne opprette en bedriftskonto	M	M	0
FK3	Kunden skal kunne opprette brukerkonto med administratorrettigheter	S	M	-1
FK4	Systemet skal verifisere registreringen	L	L	0
FK5	Dirigere opprettet bruker	S	S	0

### Innlogging

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK6	Bruker skal kunne logge inn som en brukerkonto eller bedriftskonto	XL	M	6
FK7	Systemet skal verifisere innloggingen	XL	S	7
FK8	Dirigere innlogget brukerkonto / bedriftskonto	S	S	0
FK9	Dirigere innlogget administrator	S	S	0
FK10	Bruker skal kunne tilbakesette passord	S	M	-1

### Verifiseringskode

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK11	Systemet skal verifisere bruker gjennom en kode	XL	L	4

### Presentasjon av parkeringsplasser (Feed)

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK12	Feeden skal oppliste relevante parkeringsplasser	XL	M	6
FK13	Bruker skal kunne sette et filter på feeden	L	XL	-4
FK14	Bruker skal kunne søke etter en lokasjon	M	L	-2

### Leie parkeringsplass

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK15	Bruker skal kunne leie parkeringsplass som en brukerkonto	XL	M	6
FK16	Applikasjonen skal varsle utløpstid	M	M	0

### Tilbakemelding på innlegg (Anmeldelser)

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK18	Bruker skal kunne gi tilbakemelding på et innlegg	S	L	-3
FK19	Tilbakemeldinger skal vises på hvert innlegg	S	M	-1
FK20	Utleier skal kunne få tilbakemeldinger via e-post	S	S	0

### Opprette parkeringsplasser

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK21	Brukerkonto og bedriftskonto skal kunne registrere en eller flere parkeringsplasser	XL	XL	0
FK22	Bruker skal ha tilgang til en oversikt over sine opprettede parkeringsplasser	XL	S	7

### Leiebetingelser

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK23	Pris per time skal bestemmes av utleier	M	S	1
FK24	Tiden skal bestemmes av bruker	L	M	2
FK25	Brukere skal kunne se ut ifra innlegget om parkeringsplassen er ledig eller ikke	L	M	2

### Betaling

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK26	Bruker skal betale utleieren for å opprette en reservasjon	L	M	2

## Administrering

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK27	Administrator skal kunne bannlyse bruker	S	S	0
FK28	Administrator skal kunne slette bruker	M	S	1
FK29	Administrator skal kunne slette tilbakemeldinger	S	S	0
FK30	Administrator skal kunne slette innlegg	L	S	3
FK31	Administrator skal kunne slette parkeringsplasser	L	S	3
FK32	Administrator skal kunne slette reservasjoner	XL	S	7
FK33	Bedrift skal ha administrerende rettigheter på deres egne innlegg	S	M	-1
FK34	Bruker skal kunne rapportere andre brukere sine innlegg	M	S	1
FK35	Bruker skal kunne rapportere andre brukere	M	S	1
FK36	Bruker skal kunne slette sine reservasjoner	M	L	-2
FK37	Bruker skal kunne slette sine parkeringsplasser	L	M	2
FK38	Bruker skal kunne slette sine innlegg	S	M	-1

## Persistent data

ID	Krav	Vikitighet	Utviklingstid	Størrelse
FK39	Systemet skal lagre informasjon om parkeringsplasser	XL	L	4
FK40	Systemet skal lagre informasjon om innlegg	XL	L	4
FK41	Systemet skal lagre informasjon om brukere	XL	L	4
FK42	Systemet skal lagre informasjon om reservasjoner	XL	L	4

## 14.3 Estimeringstabell - Ikke funksjonelle krav

### Tilgjengelighet

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK1	Bruker skal ha tilgang til applikasjonen	XL	XL	0
IFK2	Bruker må kvalifiseres ved tilgang til applikasjonen	L	M	2

## Sikkerhetskopi

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK3	Kunde skal ta delvis backup av database	XL	S	7
IFK4	Kunde skal ta full backup av database	XL	S	7

## Tredjepart

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK5	Betaling skal gå gjennom en tredjepart	XL	S	7

## Brukerdata og sikkerhet

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK6	Applikasjonen skal opplyse om vilkår for bruk	L	M	2
IFK7	Applikasjonen skal utdype om personvern	XL	XL	0
IFK8	Applikasjonen skal følge en standard for sikkerhet	XL	L	4

## Programvare

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK9	Tidsrom ved oppstart skal være tilstrekkelig	S	L	-3
IFK10	Tidsrom ved sidebytte skal være tilstrekkelig	S	M	-1
IFK11	Programvaren skal kunne støtte drastiske endringer	M	L	-2
IFK12	Viktige deler av programvaren skal testes	S	M	-1

## Samsvar

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK13	Parkeringsplasser må samsvare med retningslinjer	M	S	1

## Avhengigheter

ID	Krav	Vikitighet	Utviklingstid	Størrelse
IFK14	Systemet skal bruke JavaJDK 11	XL	S	7
IFK15	Systemet skal bruke Maven	L	S	3
IFK16	Systemet skal bruke Javalin	XL	S	7
IFK17	Systemet skal bruke Java JSON biblioteket Jackson	M	M	0
IFK18	Systemet skal bruke JUnit-Jupiter	L	XL	-4
IFK19	Systemet skal bruke Vue.js	XL	XL	0
IFK20	Systemet skal ha testing for web	M	M	0