

Software Engineering H2022

Gruppeoppgaven

Ashti Mawloudi

Lorena Holmeide

Rachel Medina Husanovic

Samuele Carlo Davies

Stian Karlsen Rusvik

Innholdsfortegnelse

1. Introduksjon	3
2. Mål for systemet	3
3. Målgruppen for vårt system	4
3.1 Målgruppe 1: Leietakere	4
3.1.1 Leietaker 1	4
3.1.2 Leietaker 2	6
3.1.3 Leietaker 3	8
3.2 Målgruppe 2: Utleiere	10
3.2.1 Utleier 1	10
3.2.2 Utleier 2	11
4. Løsning på problemstilling	15
5. Noen begrensninger	16
6. Krav	17
6.2 Funksjonelle krav	17
6.2 Ikke funksjonelle krav	21
7. Estimering	22
8. Videre utvikling og muligheter for systemet	28
9. Systemarkitektur	31
10. Prototypen	33
10.1 Beskrivelse av systemet	33
10.2 Avgrensninger	40
10.3 Modellering	40
10.4 Avhengigheter	50
10.5 Tester	51
10.6 Svakheter	51
11. Kildeliste	53

Prosjektdokumentasjon

1. Introduksjon

Behovet for tilgangen til en bil og det å eie en bil har endret seg fra tidligere år. For noen kan det være på grunn av et ønske om å ha en miljøvennlig atferd, mens for andre har tider med trangere økonomi som følge av f.eks. koronakrisen eller høye strømpriser endret behovet for bil. Høye kostnader ved å eie en bil eller det å ha en miljøbevisst atferd, har medført til nye måter å senke bilkostnadene og CO₂-utslippet på. Det er en tilbuds- og etterspørselsside hvor noen ønsker å leie ut sin bil, og andre som har behov for å leie en bil på enkelte dager eller i kortere perioder.

Vi ønsker å utvikle og presentere en løsning som kan tilfredsstille både de som ønsker å leie ut en bil, og de som ønsker å leie seg en bil.

Basert på dette er problemstilling vi ønsker å benytte definert slik:

- *Hvordan kan man lage en løsning der bileiere kan låne ut sine egne biler til andre som har behov for dem når de ikke selv benytter dem?*

2. Mål for systemet

Målet vårt er å ha en løsning som tilfredsstiller brukerne av applikasjonen ved å være brukervennlig, effektiv og som gir en merverdi. Med brukervennlighet mener vi at systemet skal kunne brukes av alle personer uavhengig av funksjonsevne og teknologiske evner. Det betyr at systemet skal kunne være enkelt å bruke og sette seg inn i for ulike brukergrupper. Effektivitet handler om at systemet skal holde det den lover og fungere optimalt for brukeren. Det at systemet skal gi kunden en merverdi betyr at det skal være givende å bruke systemet vi har utviklet fremfor å bruke systemer fra andre konkurrenter. For eksempel så skal systemet by på kvalitet som ligger litt over det brukeren kan forvente.

3. Målgruppen for vårt system

Målgruppen for systemet er privatpersoner fra 18 år og oppover som har førerkort på bil eller privatpersoner som har en eller flere biler som ikke blir brukt. De kan ha et ønske om å leie ut sin bil eller å leie en bil av noen for en kortere periode. Motivene for å leie en bil eller å leie ut en bil vil variere fra person til person og avhenge av vedkommendes utfordringer og behov.

For å kunne lage et system som passer ulike typer personer og brukersituasjoner, så har vi valgt å fremstille det ved bruk av personaser og tilhørende usecases. De tre første personasene er tre personer som er i ulike situasjoner, men som alle har behov for å leie seg en bil. Deretter tar vi for oss tilbudssiden, som er to personer i forskjellige situasjoner som har behov for å leie ut sine biler. Dette vil gjøre det tydelig hvilke behov de ulike brukergruppene har. Derifra kan man utvikle nødvendige funksjonelle krav som kan tilfredsstille brukerne av systemet.

3.1 Målgruppe 1: Leietakere

3.1.1 Leietaker 1

Persona

Even på 18 år går siste året på videregående skole i Fredrikstad. På 18-årsdagen sin fikk han førerkort på automat bil. Han er aktiv med fotball, og kan sykle eller gå til treninger. Annenhver helg har han derimot behov for en bil for å komme seg til bortekamper som er ulike steder på Østlandet. Han ønsker å bli mer selvstendig, og uavhengig av at foreldrene kjører han. I tillegg savner han fleksibilitet i hverdagen, som å kunne dra til kompiser som bor lenger unna.

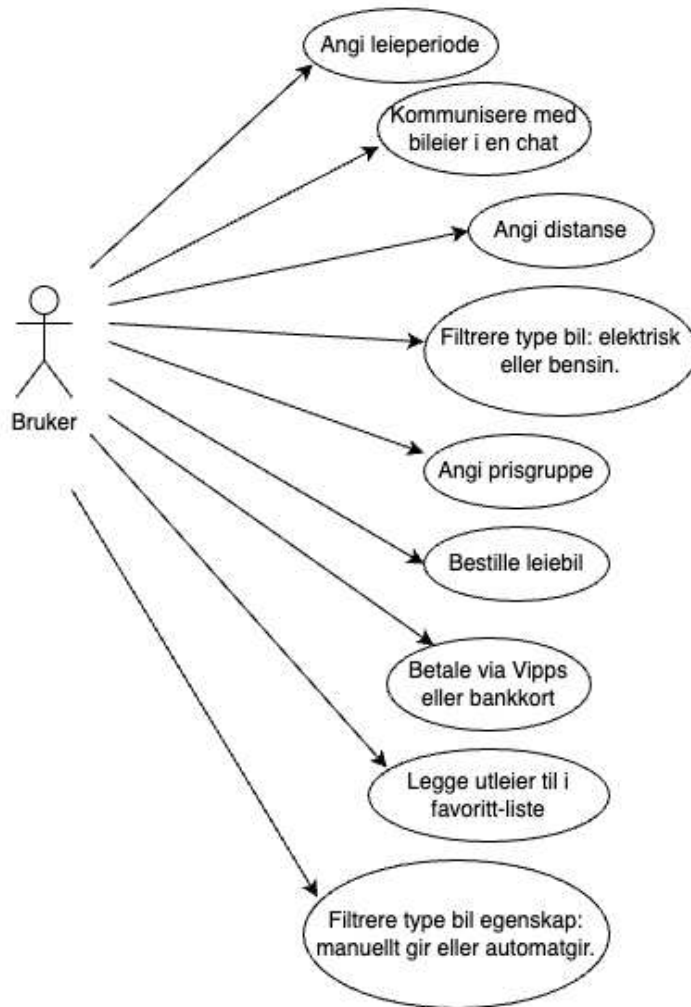
Han er på de fleste sosiale medie- plattformer som Instagram og Snapchat, og er godt vant til å bruke ulike apper i forbindelse med skolearbeid og kommunikasjon med venner. Even jobber kun i feriene som deltidsansatt på den lokale matbutikken, og har derfor ikke råd til å eie en egen bil. Han er usikker på hvor han skal studere året etter videregående, så han ønsker derfor heller ikke å gå til innkjøp av en bil.

Scenario

Even har en bortekamp kl.18:00-19:00 Torsdag ettermiddag i Moss. Det tar omtrent 40 minutter å kjøre til idrettsanlegget. Han avslutter kl.16:00 på skolen, og han ønsker å dra hjem i mellomtiden for så å reise hjemmefra kl. 17:00. Det vil være for dyrt for han å bestille taxi begge veier, samt at han ikke ønsker å vente. I tillegg er det usikkert hva han kan forvente å betale, og det er ikke alle taxisentraler som kjører denne ruten. Det vil også være tidkrevende å finne ut av hvilken taxisentral som er billigst. Even vil da ha behov for en leiebil i tidsrommet kl. 17 - 20. Han tar raskt fram mobilen fra lomma på vei hjem fra skolen, og ved bruk av applikasjonen kan han enkelt se hvilke tilbud som passer hans behov best. Bilene som er nærmest kommer opp først i listen av mulige alternativer. Han velger å filtrere bort høye prisgrupper, og legger inn et klokkeslett på når han ønsker å hente og levere fra seg bilen. Biler som er tilgjengelig i det tidsrommet blir vist.

Et problem som oppstår er at han ikke kan filtrere bort biler med manuelt gir, som vil si at han bruker litt ekstra tid på å finne en leiebil i en lav prisgruppe med automatgir. Etter at han velger ut den nærmeste automat-bilen til den laveste prisen, legger han inn informasjon til eier i et meldingsfelt om det tidsrommet han har behov for bilen. Bilen vil hentes kl.17:00 og være hos eier kl. 20:00. Han klikker seg til neste steg i prosessen som er å betale for leiebilen. Han får da valget mellom å legge inn et betalingskort eller å betale via Vipps. Han har dårlig tid, så han velger vipps og blir videreført til Vipps-appen for å betale. Når betalingen er gjennomført, får han et varsel fra applikasjonen om at den er godkjent. Når klokken er 16:50 får Even beskjed av eieren i appen i en chat-funksjon, om at bilen er klar til henting med instruksjoner på hvor nøkkelen ligger. Even går til eierens hjem som er i nærheten, og finner nøklene i postkassen som har en kode. Koden til postkassen fikk han på melding. Bilen har full tank, og han kan da kjøre til og fra Moss uten problemer.

Use Case-diagram



3.1.2 Leietaker 2

Persona

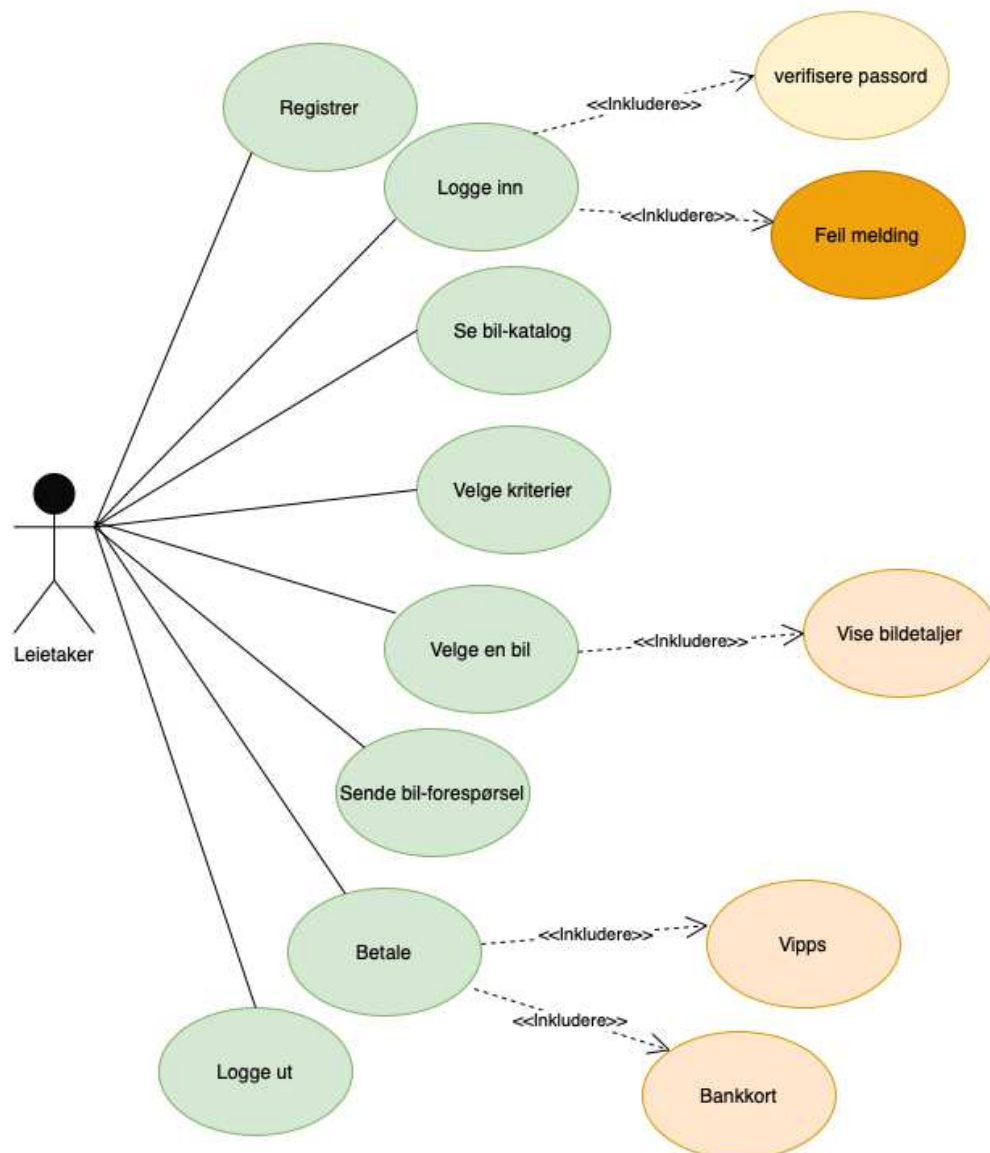
Jill er en gift kvinne på 41 år som er mor til to barn. Hun er barneskolelærer, og på fritiden liker hun å sy og løpe. I tillegg har hun hovedansvaret for sin far som bor på et gamlehjem en time unna. Hennes husholdning har kun én bil.

Jill frustrerer seg ofte over at hun ikke har tilgang til transportmidler når hennes mann er på jobb. Hun synes det tar for lang tid å benytte seg av offentlige transportmidler når hun skal besøke faren sin. Det er både kostbart og tidkrevende, spesielt når hun ønsker å ta med seg sine barn. Hun ønsker å ha flere transportmuligheter, samt å få mye mer for pengene enn det som er mulig for henne i dag. Hun drømmer om mer bekvemmelighet i forhold til transportmidler og for reiseveien til faren.

Scenario

Jill trykker seg inn på appen og registrerer seg. Deretter går hun til katalogen som viser en oversikt over ulike biler. Hun kan enten navigere seg gjennom katalogen, eller så kan hun velge ut biler basert på de kriteriene hun selv ønsker. Hun velger en bil fra listen og sender eieren en forespørsel. Etter at eieren har akseptert forespørselen, så kan hun enten betale eieren direkte med et bankkort eller betale via Vipps.

Use Case-diagram



3.1.3 Leietaker 3

Persona

Aleksander Rynning er 48 år gammel og jobber på Høgskolen i Østfold. Han er skilt og har 3 barn, og leier for tiden en leilighet ved siden av arbeidsplassen sin.

Aleksander prøver å spare penger for å kjøpe sitt eget hus. Nå kutter han ned alle de unødvendige tingene han ikke trenger til hverdagen, blant annet det å eie en bil.

Han har tre barn på alderen 7 år, 9 år og 10 år. De deltar aktivt i forskjellige sportslige klubber hver helg. Barna bor ikke sammen med ham, og han møter dem kun i helgene. Det er derfor Aleksanders ansvar å kjøre barna sine til trening, slik at de skal kunne delta på aktivitetene i tide og hente dem etter treningen.

Aleksander er ofte bekymret for om han kan klare å få hentet barna sine i tide etter jobb, og videre få levert dem tidsnok til trening. Han ønsker derfor en tjeneste-applikasjon for leiebiler rundt i nabolaget. Det kan gjøre det lettere for ham å velge en bil for å få hente barna sine, levere dem til treningsstedene deres uten stress, og unngå å komme for sent til avtaler.

Han foretrekker å leie bil i stedet for å benytte seg av offentlige transportmidler for å nå sine destinasjoner. I tillegg ønsker han å finne en bil som er i nærheten av der han bor. Han ønsker ikke avstander som krever at han må ta buss for å komme seg fram og tilbake til der utleie-bilen står. Han foretrekker også en billig leiebil, fordi han ikke har budsjett til å betale høye leiesummer. Det viktigste for han er at bilen fungerer og bringer han fra punkt A til B.

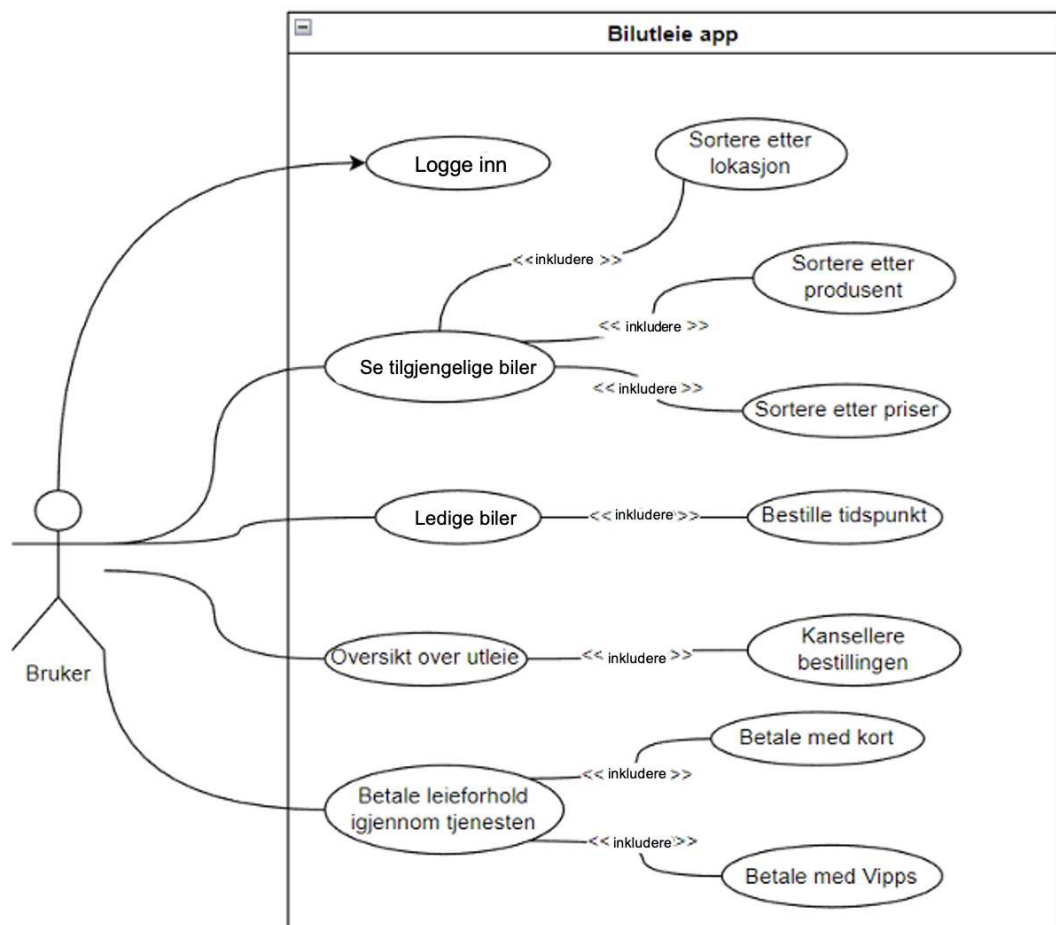
Scenario

Aleksander åpner appen gjennom en web/ mobil- applikasjon. Deretter trykker han knappen som viser "tilgjengelige biler til leie". Der vil han få valget om å filtrere etter hans egne ønsker. For eksempel å sortere etter bilprodusent, sortere pris og sortere

basert på plassering av leiebilen. Aleksander kan legge potensielle leiebiler i en egen favoritt mappe, dersom han finner en bil han ønsker å leie, men vil søke videre etter flere tilgjengelige biler på stedet. Da vil bilannonsen bli lagret i favorittlisten og han kan gå tilbake å se på annonsen på et senere tidspunkt.

Når han finner en bil han ønsker å leie, så kan han klikke på leie-knappen og gå videre til bestillings alternativene for hvilket tidspunkt han ønsker leie bilen. Deretter kan Aleksander gå videre til betalings-knappen når han vil ha muligheten til å betale for leien. Alternativene for betaling er enten å betale med et VISA kort eller med vipps. Når han har betalt for leiebilen vil han få en bekreftelse på at bestillingen var vellykket og at bilen er tilgjengelig for henting.

Use Case-diagram



3.2 Målgruppe 2: Utleiere

3.2.1 Utleier 1

Navn: Geir Hansen



(Tumisu, 2018)

Profesjonell bakgrunn

Yrke:
Fulltids eiendomsmegler
Årslønn:
874 000

Hovedmål

Bruke bilutleie appen til å leie ut bilene som er stående i garasjen mesteparten av tiden.

Brukermiljø

Lokasjon:
Kontor på jobb
Enheter:
Laptop, iPad og iPhone

Scenario

«Meste parten av året står bilene mine ubrukt i garasjen. I noen sjeldne tilfeller lånes de bort til familie og bekjente. Ideelt, burde de vært oppført på en markeds plass for utleie i periodene de ikke brukes. På denne måten kunne jeg fått noe avkastning på investeringene.»

Personlig bakgrunn

Alder: 45
Status:
Gift med to barn
Utdanning:
Bachelor i eiendomsmegling

Psykografi

Bilentusiast.
Eier av flere biler.
Inspireres av teknologiske løsninger.
Drives av å finne alternative løsninger til inntekt.

Scenario

«Min hektiske hverdag som eiendomsmegler og far til to barn, skaper en hindring for tiden jeg har til rådighet til å bruke på bilene. Av den grunn ønsker jeg en tjeneste som er enkel å bruke, hvor det skal være lett å legge inn informasjon om bilene, samt laste opp bilder. Betaling i tjenesten bør foregå sømløst, og det bør være mulig å ha oversikt over tidligere utleieforhold. I tillegg bør tjenesten inneholde et tilbakemeldingssystem, slik at man kan legge igjen anmeldelser om både leietaker og utleier. Opprettelse av brukerprofiler kan være en fordel slik at man får et inntrykk av hvem man eventuelt skal leie av, og leie ut til.»

Use Case-diagram



3.2.2 Utleier 2

Persona

Joris er en 35 år gammel mann, som er eier av en liten bedrift, i tillegg har han en kone og et barn. Opprinnelig er han fra en kystlandsby i Nord-Norge, men bor nå i en mellomstor by med en befolkning på ca. 50 000, omtrent 10 timers kjøring fra fødestedet.

Hver dag bruker Joris en sykkel eller elektrisk sparkesykkel for å reise fra hjemmet til butikken i hovedgata. På dager med tøft vær velger Joris å kjøre til jobb i sin 10 år

gamle Skoda Octavia. Joris er miljøbevisst og innser at det er bedre for miljøet å sykle, samtidig forstår han at det kan være praktisk med bil til visse tider. I noen helger og høytider drar Joris gjerne på besøk hos sin familie i Nord-Norge. Dette er en lang kjøretur, og kollektivtransport kan være svært belastende med tanke på kanselleringer, plattformbytte etc. I jule og påskeferie er tilbudet på kollektivtransport på et minimum. Barnet til Joris er mindre enn ett år, og med hensyn til dette, så er det en betryggende tanke å ha tilgang til en bil, hvis det skulle oppstå en medisinsk nødsituasjon. Disse faktorene er en av grunnene til at Joris setter pris på å ha en bil.

Bilen hans koster mye å vedlikeholde, spesielt nå som den er 10 år gammel. Han har nok penger til å dekke vedlikeholdet, men å spare noen tusen her og der hadde ikke vært feil. Joris er spesielt interessert i å potensielt kunne leie ut bilen sin i perioder, siden bilen hans står ubrukt i oppkjørselen mesteparten av uken. Sommerferien hadde også vært en ideell tid, siden været er behagelig og Joris trives med å jobbe om sommeren. Av den grunn ønsker Joris seg en løsning som gjør at han kan leie ut bilen på kort varsel, samtidig som at han ikke må forholde seg til stresset rundt betaling eller potensielle problemer som kan oppstå. En tredjepart som tar seg av dette ansvaret, men som også får en del av overskuddet, er noe som virker rimelig for Joris.

Scenario

Joris og familien har bestilt ferie for uken. De skal reise utenlands til Tsjad via flytransport. Joris innser at han bør leie ut bilen sin i perioden de er på ferie, siden ferien kommer til å koste mye penger. Dette er et mye bedre alternativt enn at den skal stå i oppkjørselen. Joris bestemmer seg derfor for å legge den ut til leie. Han åpner applikasjonen via en nettleser. Deretter trykker han på knappen “registrer bruker” for å opprette en brukerkonto. Videre blir han bedt om å verifisere personnummer via en to-faktor-autentisering, koblet gjennom en tredjeparts autentiseringstjeneste. Joris blir deretter bedt om å lese gjennom bruksvilkårene, og godta disse. Deretter klikker han på en annen knapp i navigasjonsmenyen og blir bedt om å skrive inn registreringsnummeret på bilen sin. Programmet henter så all informasjon knyttet til hans bilmodell via en ekstern tjeneste. Joris laster så opp flere bilder av bilen sin, og krysser av på funksjoner som bilen har. Bilen kan feste en standard tilhenger under

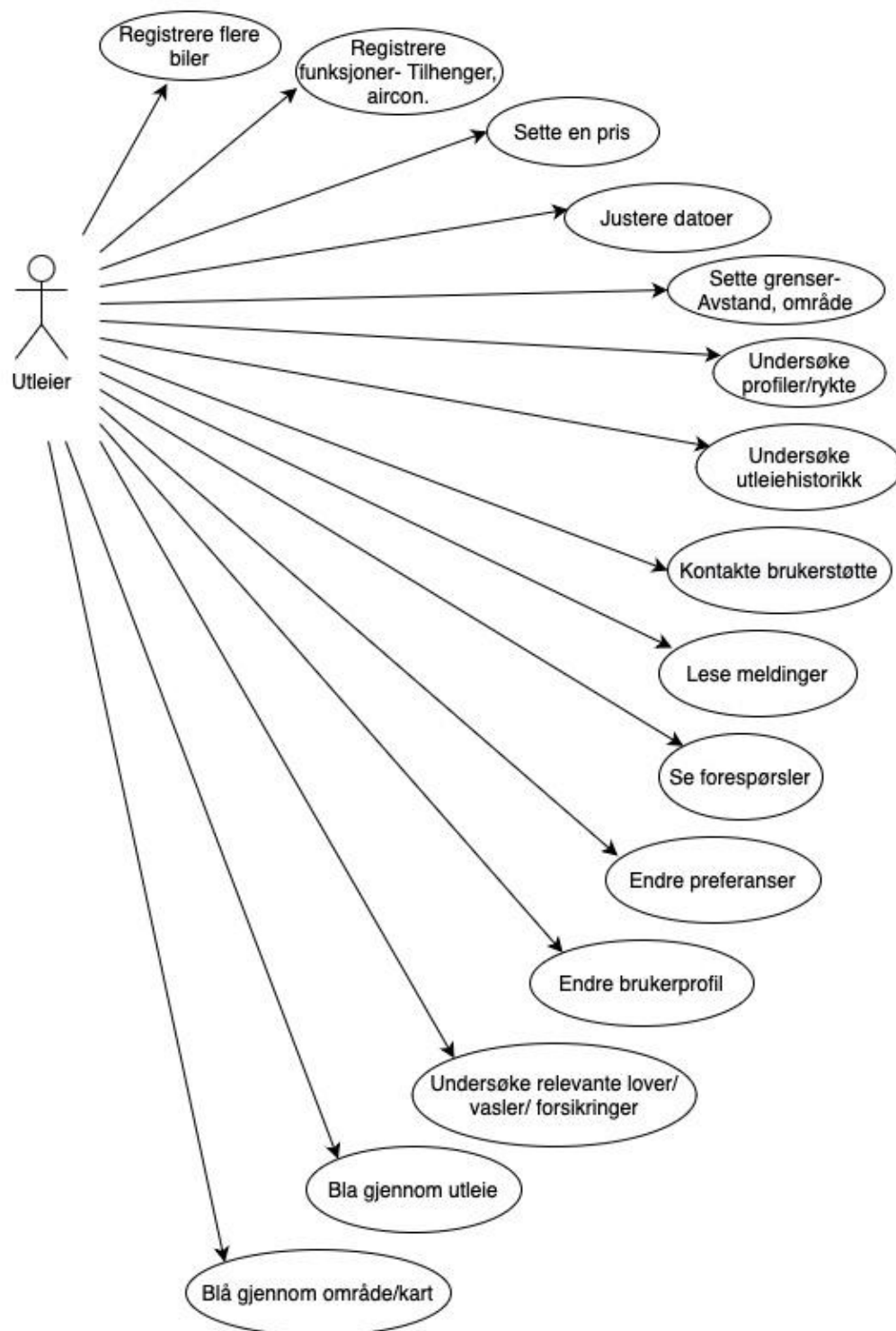
3500-kg, den har klimaanlegg, takfeste, vinterdekk, samt Bluetooth, stereoanlegg og soltak.

Etter registrering fortsetter Joris med å bla gjennom utleie i området sitt, han filtrerer på bilprodusent og årsmodell for å få et grovt estimat på hvordan han kan prise bilen sin. Joris klikker deretter på navigasjonsmenyen, og så klikker han på enda en knapp som lar han velge hvilke av bilene han ønsker å gjøre tilgjengelig for utleie. Han klikker på Skodaen sin og det dukker opp en kalender. Joris velger så datoen familien er på ferie. Ettersom at dette er første gangen Joris leier ut bilen sin, så blir han bedt om å registrere bankkontonummer, PayPal eller Vipps.

Joris har ikke hørt mye rundt bilannonsen han la ut de siste ukene, men etter hvert som ferien nærmer seg, så dukker det opp et e-postvarsel som informerer han om en innkommende melding. Meldingen er ikke en forespørsel på å leie bilen hans i ferieperioden, men en bruker som spør om han er fleksibel på datoene som han har satt for utleie. Joris spør så hvilke datoer den andre brukeren har i tankene, og går med på en avtale om å leie ut bilen sin én dag tidligere enn det han opprinnelig hadde satt i annonsen. Leien er offisielt bekreftet, og en tredjepart tar ansvar for at betaling og forsikrings håndtering går sømløst for seg.

På leveringsdagen blir bilen overlevert til leietaker, og Joris informerer om at han kan levere både bilen og nøklene hos hans bror når den tid kommer. Leietakeren bekrefter at han er fornøyd med tilstanden til bilen. Joris kommer tilbake fra ferie og oppdager at leietaker har glemt å fylle bensin, og tar dermed et bilde av drivstoffmåleren, samt kilometerstand. Bildet blir deretter arkivert sammen med en kopi av leietakerens bilde av de samme instrumentene når han leverte bilen. Joris ber om ytterligere 300-kroner for å dekke denne kostnaden. Tredjeparts-betalingstjenesten sender de pengene når leieperioden er over og alle er fornøyd.

Use case-diagram



4. Løsning på problemstilling

Løsningen vår er å tilby en applikasjon som er gunstig tilpasset for både leiere og utleiere. For å få til dette må man kontinuerlig jobbe ut ifra de mulige personene, og ta hensyn til deres scenarier, behov og utfordringer. En slik framtoning sørger for at produktet blir tilpasset mangfoldet. Vi kan starte med å ta en titt på det vi mener er de viktigste funksjonene som må være på plass for at systemet kan bli realisert og levere verdi til kunden.

Spesifikt for utleier:

- Opprette utleie annonser for biler
 - Legge til bilder
 - Beskrivelse av bilen
 - Sette ønsket utleiepris
 - Tilføre dato for tilgjengelighet
- Svare på forespørsel fra leietakere
 - Inkluderer varsling til utleier

Spesifikt for leietaker:

- Opprette en brukerprofil
 - Legge til betalingsinformasjon
 - Legge til førerkort
- Finne tilgjengelige biler i område basert på lokasjonssøk
 - Utføre filtrering blant søk
- Sende forespørsel på biler via knappetrykk
- Betale for leieperiode til utleier

Det er såklart flere faktorer som inngår for at disse punktene skal fungere sømløst i tjenesten, og en oversikt over de mer detaljerte og tekniske bitene finner man i usecase- diagrammene og i krav-tabellene. Punktene som er oppført er altså de helt essensielle funksjonene som må være på plass for at en utleier skal kunne få leid ut en bil, og at en leietaker tilsvarende skal kunne få leid seg en bil.

5. Noen begrensninger

For å øke kvaliteten på produktet må man også implementert andre viktige elementer som er av en mindre synlige art, men som sørger for bedre kundetilfredshet. Her snakkes det selvfølgelig om persistent lagring. Skal opplevelsen av å bruke applikasjonen være minneverdig, så er det helt fundamentalt at alt av annonser, interaksjoner, historikk og dialoger blir lagret permanent. På denne måten vil brukerne få tilgang til å fortsette der de slapp når de vender tilbake til applikasjonen. Siden dette er en prototype, så har vi valgt å rette fokuset mot å lagre alt av informasjon i en SQLite database (en database som er lagret i en enkelt fil). I en reell situasjon ville innføring av SQL eller NoSQL følgelig vært bedre egnet for et slikt formål.

Når det snakkes om lagring av persondata så er veien til personvern og kryptering kort. Dette må være på plass for at brukerne skal føle på en trygghet ved å dele personopplysninger. Igjen må vi ta til betraktning at dette er en prototype og at innføring av disse punktene ville vært svært tidkrevende. Vi velger derfor å luften tanken om at hvis dette hadde vært et ekte produkt som skulle blitt satt i produksjon, så ville vi lagt ned mye ressurser i å få orden på dette.

Enkelhet er noe som skaper en form for verdi til de fleste, men hva som defineres som enkelt er opp til enhver og avgjøre. Av den grunn bør det implementeres alternative metoder til å utføre søk på lokasjoner og selvfølgelig latt brukerne få teste dette for tilbakemelding. Ideelt burde det vært mulig å velge en by ved å klikke på et interaktivt kart eller benytte et input felt til å søke på gatenavn, by og postnummer. Vi valg å fokusere på sistnevnte, siden førstnevnte metode viste seg å være mye arbeid å implementere. Den siste løsningen gir fortsatt personer med et midlertidig leiebehov muligheten til å søke etter tilgjengelige biler i områder hvor de kunne tenke seg å leie et kjøretøy.

Betaling i tjenesten er noe som bør håndteres av en eller flere tredjepartsleverandører i en reell applikasjon. Siden dette er en prototype har vi kommet fram til en løsning om å implementere ulike “stubs” for betaling via forskjellige betalingsleverandører. For å dekke flest mulige brukere har vi valgt å tilby følgende metoder.

Alternative betalingsmetoder

- Vipps
- Google pay
- Paypal
- Kredittkort

Uttrykket “stub” kan tolkes som en erstatning for en avhengighet eller en samarbeidspartner. I prosjektet bruker vi det som en erstatning for tredjepartsleverandører av betalingstjenester. På denne måten kan vi skrive kode som imiterer adferden til de ulike betalingstjenestene, uten å være i direkte kontakt med tredjepartsleverandørene. I tillegg skaper dette muligheten til å teste koden som blir skrevet, slik at man kan verifisere at den fungerer som tiltenkt.

Denne løsningen tar hensyn til begge parter som er definert i problemstillingen. Ved å innføre funksjonene som har blitt nevnt ovenfor, vil man ende opp med et enkelt system som tar for seg det mest essensielle i problemstillingen. Ved implementasjon betyr dette at en bruker skal både kunne leie ut en bil, og tilsvarende leie en bil.

6. Krav

Basert på tidligere beskrevet usecase-diagrammer, persona og scenarioer (del 3) kan vi identifisere kravene systemet trenger for at det skal gi verdi til kunden og brukerne. Kravene er gruppert inn i funksjonelle og ikke-funksjonelle krav. Der de funksjonelle kravene handler om hvordan systemet skal kunne tilby brukerne ulike funksjonelle tjenester, mens de ikke-funksjonelle er krav som angår egenskapene til systemet som brukeren ikke selv benytter seg aktivt av. Kravene er kategorisert etter hvilken del av systemet funksjonene befinner seg.

6.2 Funksjonelle krav

Betaling:

1. Leietaker skal kunne betale via online betaling:

- 1.1) Paypal
- 1.2) Vipps

- 1.3) Kreditt-/Debetkort
 - 1.3.1) Visa
 - 1.3.2) Mastercard
 - 1.3.3) American Express
- 1.4) Google Pay, Apple Pay

Test: [test/java/no/hiof/groupproject/tools/VerifyPaymentTest.java](#)

Filtrering:

2. Leietaker skal kunne sortere og filtrere søkeresultater som

- 2.1) Automat eller manuelt gir
- 2.2) Diesel, bensin eller elektrisk
- 2.3) GPS
- 2.4) Bilmerke
- 2.5) Prisklasse (både daglig og per kilometer kostnader)
- 2.6) År
- 2.7) Sitteplass
- 2.8) Rykte
- 2.9) Antallet resultater

Test: [test/java/no/hiof/groupproject/tools/db/DatabaseFilteringTest.java](#)

Geografisk område:

3. Leietaker skal kunne velge geografisk område basert på å

- 3.1) Angi by

Test: [test/java/no/hiof/groupproject/tools/LocationTest.java](#)

Leieperiode:

4. Leietaker skal kunne velge dato

- 4.1) Utleier skal kunne velge tidsperioder når kjøretøyet er tilgjengelig

Test:

[test/java/no/hiof/groupproject/models/AddBookingAndAvailableWithinTest.java](#)

og [test/java/no/hiof/groupproject/tools/db/DatabasePersistenceTest.java](#)

5. Leietaker skal kunne be om kansellering

- 5.1) Leietaker skal kunne betale depositum nærme leiedato

Test: [test/java/no/hiof/groupproject/models/RemoveBookingTest.java](#)

6. Leietaker skal kunne legge inn en forespørsel om å leie en bil uten spesifikt å velge en eksisterende annonse.

Registrere kjøretøy:

7. Utleier skal kunne registrere mer enn ett kjøretøy.

Test: [test/java/no/hiof/groupproject/models/MultipleVehiclesTest.java](#)

Datainnsamling:

8. Leietaker og utleier (brukere av systemet) skal kunne etterspørre GDPR data fra administrator/firmaet av systemet.

Test: [test/java/no/hiof/groupproject/tools/GDPRInformationTest.java](#),
[test/java/no/hiof/groupproject/tools/db/DatabaseTest.java](#) og
[test/java/no/hiof/groupproject/tools/db/DatabasePersistenceTest.java](#)

Profil:

9. Leietaker og utleier skal kunne redigere egen profil

9.1) Kontaktinformasjon

9.2) Slette kjøretøy

9.3) Slette brukerkonto

Test: [test/java/no/hiof/groupproject/models/UpdateUserTest.java](#) og
[test/java/no/hiof/groupproject/models/RemoveUserTest.java](#) og
[test/java/no/hiof/groupproject/models/RemoveVehicleTest.java](#)

10. Leietaker og utleier (brukere av systemet) skal kunne se andres profiler

10.1) De skal kunne se rykte

10.1.1) De skal kunne se anmeldelser av leietakeren og utleier.

Test: [test/java/no/hiof/groupproject/models/RatingsTest.java](#)

Dokumentasjon:

11. Leietaker og utleier skal kunne arkivere bilder for dokumentasjon

11.1) Potensielle skader uhell

11.2) Km mål før/etter leieperiode

Kommunikasjon:

12. Leietaker og utleier skal kunne trykke på en knapp for å sende en melding eller forespørsel.

12.1) IM funksjon mellom to personer

Test: [test/java/no/hiof/groupproject/tools/chat/ChatRoomTest.java](#)

Registrering:

13. Systemet skal ha en registreringsside for sine brukere med vilkår fra administrator/firma(vår kunde)

13.1) Utleier og leietaker skal kunne registrere seg med en Google-konto eller Facebook-konto.

Vilkår:

14. Administrator skal oppgi vilkår for bruk av siden

14.1) Forventninger (Rengjøring, vask, påfyll av drivstoff/lading)

14.2) Depositum

14.3) Rapportere skader

14.4) To forskjellige side for brukervilkår –En for leietaker, og en for utleier.

Verifisering:

15. Systemet skal kunne verifisere forsikring og førerkort fra utleier og leietaker via en tredjepart.

Test:

- [test/java/no/hiof/groupproject/tools/VerifyInsuranceTest.java](#)
- [test/java/no/hiof/groupproject/tools/LicenseTest.java](#)

Varsler:

16. Systemet skal kunne sendes ut varsler til både GUI-et og også på e-post til brukerne.

Test: [test/java/no/hiof/groupproject/tools/NotificationTest.java](#)

6.2 Ikke funksjonelle krav

Autolagring:

17. Systemet skal ha en autolagrings-funksjon:

17.1) Sendte meldinger- lagres umiddelbart.

Test: [test/java/no/hiof/groupproject/tools/chat/ChatRoomTest.java](#)

17.2) Redigere brukerprofil – også lagret.

17.3) Transaksjoner

17.4) Lagre annonser (med AvailableWithin tidsperioder), bestillinger, lisenser, betalinger, brukere, kjøretøy, og brukerprofiler (med vurderinger).

Test: [test/java/no/hiof/groupproject/tools/db/DatabasePersistenceTest.java](#)

GDPR:

18. Systemet skal følge GDPR

18.1) Samtykkeskjemaer og avtaler

18.1.1) Kan fortsatt bli gjennom tjenesten hvis avtalen blir avvist av bruker.

18.2) Sikkerhet

18.2.1) Systemet skal ha tofaktor-autentisering ved innlogging og betaling

18.3) Kryptering av sensitive data fra brukerne.

18.4) Informasjon som ikke trenger å bli lagret skal slettes.

18.4.1) Systemet skal ta hensyn til personvern.

Lokalisering/plassering:

19. Søkeresultatene fra leietaker skal kunne sorteres basert på rykter.

Test: [test/java/no/hiof/groupproject/tools/db/DatabaseFilteringTest.java](#)

20. Brukeren bør bli bedt om å få tilgang til plassering når man foretar en forespørsel.

20.1) Hvis den avvises, skal standardplasseringen brukes.

20.2) Basert på rykter

Test: [test/java/no/hiof/groupproject/tools/db/DatabaseFilteringTest.java](#)

Frister:

21. Utleier skal måtte forholde seg til frister for å akseptere/avslå et tilbud fra leietaker.

21.1) Leietaker skal kunne få svar innen 48 timer.

Test: [test/java/no/hiof/groupproject/tools/DeadlineTest.java](#)

Betaling:

22. Systemet skal spørre om autentisering når leietaker betaler

22.1) Sesjons ID og informasjonskapsler

Test: [test/java/no/hiof/groupproject/tools/VerifyPaymentTest.java](#)

Søkeresultater:

23. Systemet skal oppdatere søkeresultater for hvert søk for å inkludere nye resultater.

24. Systemet skal kunne sortere søkeresultater basert på omdømme til utleier.

Test: [test/java/no/hiof/groupproject/tools/db/DatabaseFilteringTest.java](#)

Oppetid:

25. Systemet skal ha en oppetid lik 24/7.

26. Systemet skal kunne støtte flere forespørsler samtidig.

Utforming:

27. Universell utforming og WCAG

27.1) Forpliktet til å bestå 35 av 61 krv

27.2) Mange av disse er UI-basert

7. Estimering

De funksjonelle og ikke funksjonelle kravene er satt inn i en tabell som grovestimerer estimatenes vanskelighet, viktighet og prioritet. Estimatenes Vanskelighet (utviklingstid) handler om hvor omfattende vi vurderer at et krav vil være å implementere i prototypen. Med andre ord, hvor tidkrevende vi anser det er å implementere kravet.

Viktighet handler om graden av hvor viktig det er å ha med den enkelte funksjonen for at applikasjonen skal levere den verdien kunden forventer. Estimatet Prioritet handler om hvordan en funksjons viktighet står i forhold til de andre funksjonene. Det viser dermed hvilke av kravene som bør implementeres før de andre.

Tabellen har ulike verdier basert på t-skjorte størrelser. S er rangert som en lav verdi, M som middels verdi, L som stor verdi og XL er ansett som en svært høy verdi. For prioritert er det tallverdier fra negativ til positiv som gir en prioriteringsrekkefølge basert på hva som gir størst nytte hos kunden. Desto høyere positiv verdi, jo høyere opp i prioriteringen anser vi kravet til å være. Høy prioritet er basert på at kravet har en lav vanskelighet for implementasjon og en høy viktighet.

Tabellen gir en oversikt over hva som er viktig og hva som er vanskelig, samt hva som bør prioriteres og hvordan. Prototypen er derfor basert på denne estimeringen, hvor estimatet Prioritering har en betydning for hvilke krav vi har implementert.

Tabellen under viser tallverdiene til de enkelte størrelsene:

	Utviklingsstørrelse (vanskelighet)			
Verdi (viktighet)	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

Tabellen viser de funksjonelle kravene sortert etter estimatet Prioritet med tilhørende tallverdier:

<u>Funksjonelle krav</u>	Utviklingstid (Vanskelighet)	Viktighet	Prioritet (Nettoverdi)
1. Leietaker skal kunne betale via online betaling: 1.1) Paypal 1.2) Vipps 1.3) Kreditt-/Debetkort 1.3.1) Visa 1.3.2) Mastercard 1.3.3) American Express 1.4) Google Pay, Apple Pay	S	XL	7
2. Leietaker skal kunne sortere og filtrere søkeresultater som 2.1) Automat eller manuelt gir 2.2) Diesel, bensin eller elektrisk 2.3) GPS 2.4) Bilmerke 2.5) Prisklasse (både daglig og per kilometer kostnader) 2.6) År 2.7) Sitteplass 2.8) Rykte 2.9) Antall resultater	M	XL	6
15. Systemet skal kunne verifisere forsikring og førerkort fra utleier og leietaker via en tredjepart.	S	L	3
5. Leietaker skal kunne be om kansellering 5.1) Leietaker skal kunne betale depositum nærme leiedato.	M	L	2

12. Leietaker og utleier skal kunne trykke på en knapp for å sende en melding eller forespørsel. 12.1) IM funksjon mellom to personer	M	L	2
13. Systemet skal ha en registreringsside for sine brukere med vilkår fra administrator/firma(vår kunde). 13.1) Utleier og leietaker skal kunne registrere seg med en Google-konto eller Facebook-konto.	M	L	2
9. Leietaker og utleier skal kunne redigere egen profil 9.1) Legge inn anmeldelser 9.2) Kontaktinformasjon 9.3) Slette kjøretøy 9.4) Slette brukerkonto	M	L	2
16. Systemet skal kunne sendes ut varsler til både GUI-et og også på e-post til brukerne.	S	M	1
7. Utleier skal kunne registrere mer enn ett kjøretøy	S	M	1
14. Administrator skal oppgi vilkår for bruk av siden 14.1) Forventninger (Rengjøring, vask, påfyll av drivstoff/lading) 14.2) Depositum 14.3) Rapportere skader 14.4) To forskjellige side for brukervilkår –En for leietaker, og en for utleier.	S	M	1
3. Leietaker skal kunne velge geografisk område basert på å: 3.1) Angi by	L	L	0

11. Leietaker og utleier skal kunne arkivere bilder for dokumentasjon 11.1) Potensielle skader uhell 11.2) Km mål før/etter leieperiode	L	L	0
6. Leietaker skal kunne legge inn en forespørsel om å leie en bil uten spesifikt å velge en eksisterende annonse.	M	M	0
4. Leietaker skal kunne velge dato	M	M	0
10. Leietaker og utleier (brukere av systemet) skal kunne se andres profiler 10.1) De skal kunne se rykte. 10.1.1) De skal kunne se anmeldelser av leietakeren, og utleier.	M	M	0
8. Leietaker og Utleier (brukere av systemet) skal kunne etterspørre GDPR data fra administrator/firmaet av systemet.	M	S	-1

Tabellen under viser de ikke- funksjonelle kravene sortert etter estimatet Prioritet med tilhørende tallverdier:

<u>Ikke-funksjonelle krav</u>	Utviklingstid (Vanskelighet)	Viktighet	Prioritet (Nettoverdi)
17. Systemet skal ha en autolagrings-funksjon: 17.1) Sendte meldinger- lagres umiddelbart. 17.2) Redigere brukerprofil – også lagret. 17.3) Transaksjoner 17.4) Lagre annonser (med AvailableWithin tidsperioder), bestillinger, lisenser, betalinger,	L	XL	4

brukere, kjøretøy, og brukerprofiler(med vurderinger).			
26. Systemet skal kunne støtte flere forespørsler samtidig.	S	L	3
21. Utleier skal måtte forholde seg til frister for å akseptere/avslå et tilbud fra leietaker. 21.1) Leietaker skal kunne få svar innen 48 timer.	S	L	3
19. Søkeresultatene fra leietaker skal kunne sorteres basert på rykter.	M	L	2
22. Systemet skal spørre om autentisering når leietaker betaler 22.1) Sesjons ID og informasjonskapsler	M	L	2
24. Systemet skal kunne sortere søkeresultater basert på omdømme til utleier.	S	M	1
23. Systemet skal oppdatere søkeresultater for hvert søk for å inkludere nye resultater.	M	M	0
25. Systemet skal ha en oppetid lik 24/7.	S	S	0
20. Brukeren bør bli bedt om å få tilgang til plassering når man foretar en forespørsel. 20.1) Hvis den avvises, skal standardplasseringen brukes. 20.2) Basert på profil.	S	S	0
27. Universell utforming og WCAG 27.1) Forpliktet til å bestå 35 av 61 krav 27.2) Mange av disse er UI-basert	M	S	-1
18. Systemet skal følge GDPR 18.1) Samtykkeskjemaer og avtaler 18.1.1) Kan fortsatt bla gjennom tjenesten hvis avtalen	L	M	-2

blir avvist av bruker. 18.2) Sikkerhet 18.2.1) Systemet skal ha tofaktor- autentisering ved innlogging og betaling. 18.3) Kryptering av sensitive data fra brukerne. 18.4) Informasjon som ikke trenger å bli lagret skal slettes. 18.4.1) Systemet skal ta hensyn til personvern.			
--	--	--	--

Estimeringen viser at det er kravene 1, 2, 15 og 18 som gir høyest prioritering basert på hva brukeren vil oppleve å få høyest verdi av. Dette har vi tatt hensyn til under utviklingen av prototypen. Det betyr at vi har prioritert disse kravene først, før vi har valgt å implementere de resterende som vi ser at gir en verdi til brukeren. Krav relatert til GDPR og universell utforming er nedprioritert i prototypen. Det tilsier ikke at det ikke er viktig for systemet som tar opp mye personlig informasjon og data av brukerne, samt at det bør være brukervennlig med tanke på ulike funksjonsevne hos ulike brukerne.

8. Videre utvikling og muligheter for systemet

Prototypen inneholder de mest vesentlige og viktigste tingene vi mener gir verdi til kunden og brukerne av systemet. Det er dermed ikke slik at prototypen er et ferdig produkt. Vi har gjort oss opp noen refleksjoner rundt systemet, og mener at prototypen har et potensiale for å bli videreutviklet. Vi ville blant annet ha implementert flere funksjoner, dersom vi hadde hatt muligheten til å utvikle et ferdig produkt.

Krav 3 kan for eksempel ha en tilleggsfunksjon, hvor brukeren avgrensar radiusen i km på et kart som viser biler som er mulig å leie. Dette er en stor fordel for leietakeren når man ønsker visuelt å se en god og rask oversikt over hvilke ledige biler som er i nærheten. Krav 4 har også mulighet for en tilleggsfunksjon der

leietakere kan sende inn bud og mottilbud. Det er gunstig for utleier med tanke på økt profitt, samt for leietaker som kan være med å påvirke muligheten til å få en bestemt ønsket bil for en angitt periode.

Vi ser også muligheter til å forbedre krav 9, 10, 12 og 17 med tilleggsfunksjoner.

De fleste applikasjoner med mulighet for kommunikasjon mellom to eller flere parter har som oftest en introduksjon av partene i form av en kort biografi i profilen(bio).

Krav 9 har potensialet til å legge til det som en ekstra funksjon. Det vil gi leietaker og utleier en mulighet til å få en presentasjon av hvem det er de kommuniserer med, og kan styrke relasjonen og troverdigheten mellom dem. Det kan gjøre det lettere for utleier å godta en forespørsel, og det kan gjøre det lettere for leietaker å foreta en forespørsel.

På krav nummer 10 har vi satt at brukerne skal kunne se rykte til hverandre, men det kan også være en fordel å kunne se og anmelde med kommentarer for å øke tilliten mellom utleier og leietaker. For utleier vil det være nyttig å kunne se hva som f.eks. var bra for leietakeren, og eventuelt lese hva han kan gjøre bedre for neste leietaker. Det samme kan gjelde for en kjøretøyet til utleier, samt leietakeren. Det er minst like viktig for utleieren å kunne stole på at leietakeren behandler kjøretøyet med måte. En kort kommentar fra tidligere utleiere på leietaker vil kunne være betryggende for utleieren om at leietaker er pålitelig. Leietakerne kan være nysgjerrige på opplevelsen av kjøretøyet, og da vil det være til hjelp med en funksjon som gir leietakerne mulighet til å kommentere selve kjøretøyet til utleier.

Kommunikasjonen mellom brukerne via meldingsfunksjonen i krav 12 gir leietaker og utleier en rask måte å kommunisere med hverandre på. Det kan dog oppleves for noen utleiere eller leietakere som et forstyrrelsesmoment i hverdagen, dersom det blir unødvendig mye spørsmål eller at man havner ovenfor en uoverensstemmelse av f.eks. bruken av kjøretøyet. Derfor kan det være nødvendig med en tilleggsfunksjon som gir mulighet til å blokkere kommunikasjonen i meldingsfunksjonen.

Krav 17 gir rom for en ytterligere funksjon som lagrer automatisk favoriserte kjøretøy eller bokmerke av et kjøretøy eller en profil. Da vil leietaker ikke trenge å huske en spesifikk leietaker, men kan da ha utleieren lagret på sin brukerkonto. Nært dette vil det være mulighet for å lage en funksjon som lagrer favoriserte profiler av utleiere i en egen favorittliste. Det vil gjøre det enkelt for en leietaker å samle potensielle utleiere.

Krav nr. 6 er ikke implementert i prototypen, men det vil være tilfeller der leietakere vet i god tid i forkant at de trenger en bil for en gitt periode. En funksjon som lar leietakerne opprette en annonse som forespør om å leie en bil uten å velge en spesifikk leietakers annonse, vil kunne gi verdi til leietakeren som da unngår å bruke tid på å finne en utleier. Det vil da være leietaker som bestemmer om han ønsker å benytte seg av tilbudet en utleier gir som følge av forespørsels-annonsen.

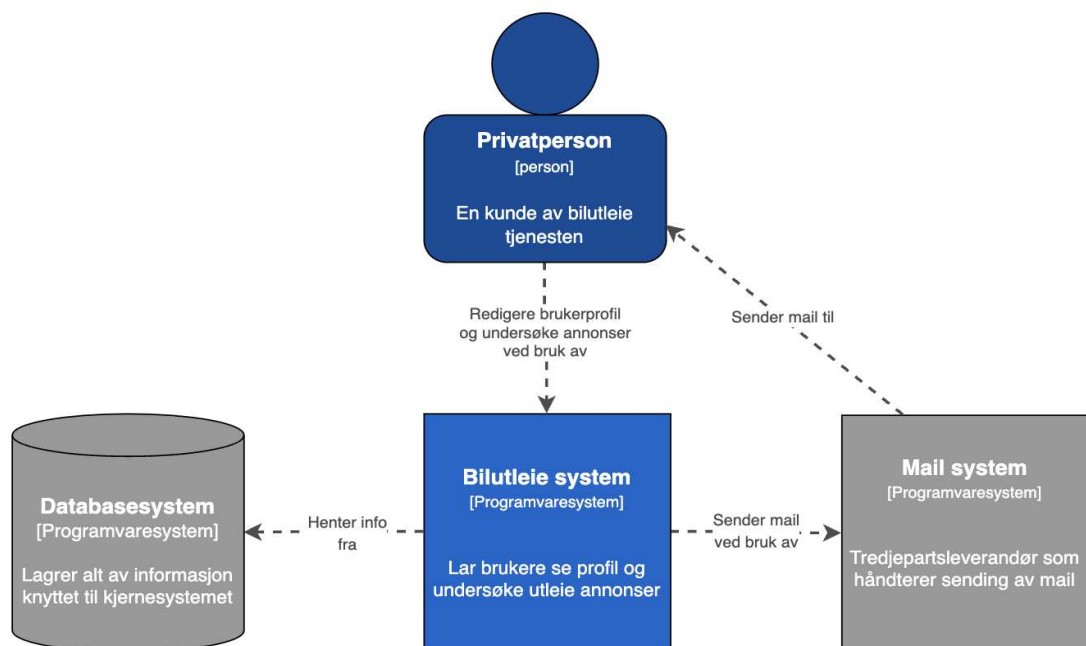
Krav 11 har også et potensiale for å bli implementert i fremtiden, siden det vil være lurt for både leietaker og utleier å ha en oversikt over mulige skader som kan forekomme under utleieperioden. Ved å kunne henvise til bilder som er lagret i systemet, vil de kunne se hvilke skader som er skjedd på hvilke tidspunkter og av hvem. Det er også en fordel å kunne lagre bilder som viser ant kilometer som bilen er målt til å ha kjørt under leieperioden. Det vil påvirke verdien til den utleide bilen.

Krav 27 er blitt nedprioritert i prototypen, og er noe vi ser for oss at kan bli videreutviklet i fremtiden. Det er viktig å ta hensyn til universell utforming og WCAG for at brukerne alle kan bruke systemet uavhengig av deres funksjonsevne til å navigere seg rundt i applikasjonen. Innholdet må være tydelig og logisk for brukeren. WCAG 2.0-standardens krever at 35 av 61 kriterier må være oppfylt (Uutilsynet, u.å.). En annen funksjon som også vil påvirke brukervennligheten er å ha en applikasjon som er tilgjengelig på både pc og mobil-enheter, og er dermed nettstedbasert. I tillegg vil en kundeservice side for utleier og leietakere gjøre applikasjonen mer brukervennlig, ettersom at man da vil kunne få hjelp med problemer man støter på som bruker.

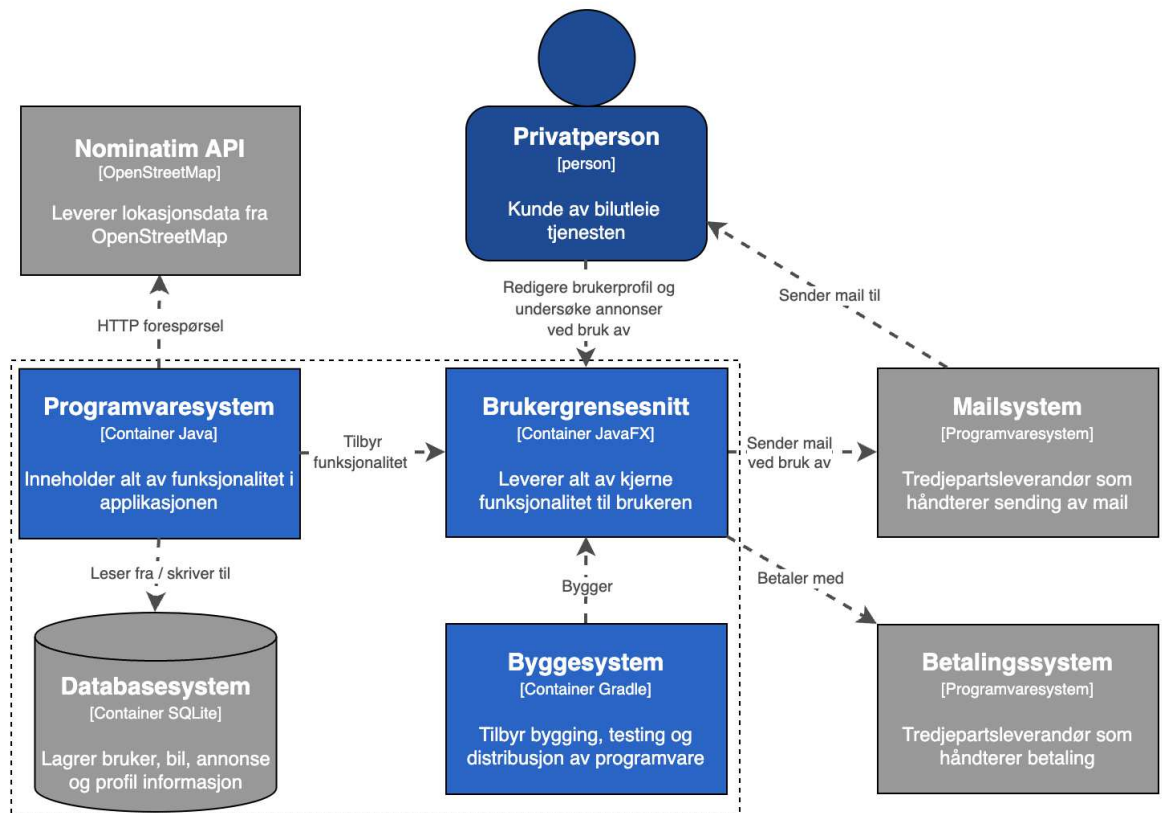
9. Systemarkitektur

Arkitekturen er et løsningsforslag til hvordan et tiltenkt system kan se ut basert på opplysningene i oppgavebeskrivelsen. Vi har valgt å benytte oss av C4-modellen, en modell som har blitt utviklet for å hjelpe utviklerteam med å forklare og kommunisere programvarearkitektur. Det er en måte å lage et “kart” over programvaren av ulikt detaljnivå, som ofte inneholder fire ulike nivåer. Her har vi valgt å fokusere på de to første, da de to siste kan bli litt vel tekniske.

Modellen implementerer zoom-egenskapene til et interaktivt kart ved at diagrammene blir mer detaljert jo lenger ned man beveger seg i nivåene. Den viser den overordnede strukturen i systemet og sammenhengen mellom de ulike delene system er bygget av.



Nivå 1: Et kontekstdiagram som viser hvordan applikasjonen passer inn i verden rundt.



Nivå 2: Et kontainer diagram som viser sammenhengen mellom komponentene.

Diagrammene gir en oversikt over teknologi, rammeverk og tredjepartsleverandører som kan bli benyttet i et slikt system. Her er det viktig å bemerke seg at tredjepartsleverandørene ikke er implementert, men at det heller har blitt skrevet “stubs” for å imitere lignende funksjonalitet. Det som faktisk blir benyttet i prototypen er følgende punkter:

- Programmeringsspråk: Java
- Rammeverk for brukergrensesnitt: JavaFx
- Byggesystem: Gradle
- Databasesystem: SQLite
- Programvare for geokoding: Nominatim API

Komponentene i den rektangulære boksen representerer selve kjernen av systemet, mens alle komponentene som befinner seg utenfor boksen er elementer som systemet kommuniserer med.

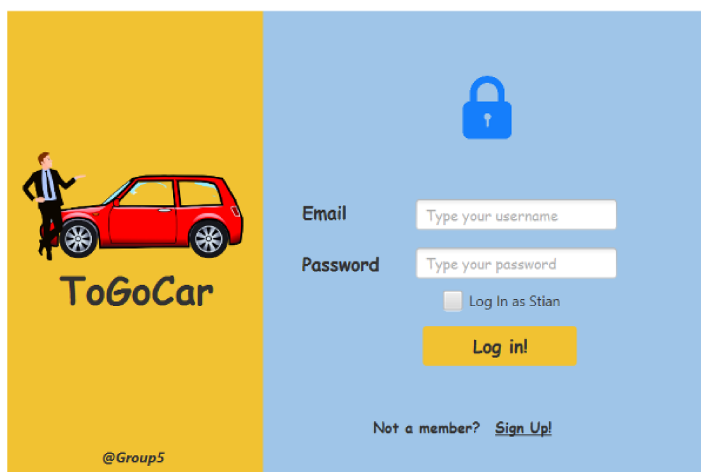
10. Prototypen

10.1 Beskrivelse av systemet

Prototypen inneholder de funksjonene som vi anser som viktige for at systemet skal kunne gi verdi til kunden, og derav brukerne. For å laste ned prosjektet i kommandolinjen skrives: `git clone https://github.com/stian96/Group_Project.git`. Man må ha lastet ned JAVA 17 for at applikasjonen skal fungere, samt Gradle v7.2. For at GUI-et skal komme opp fra kommandolinjen må følgende kommando brukes: `gradlew run`

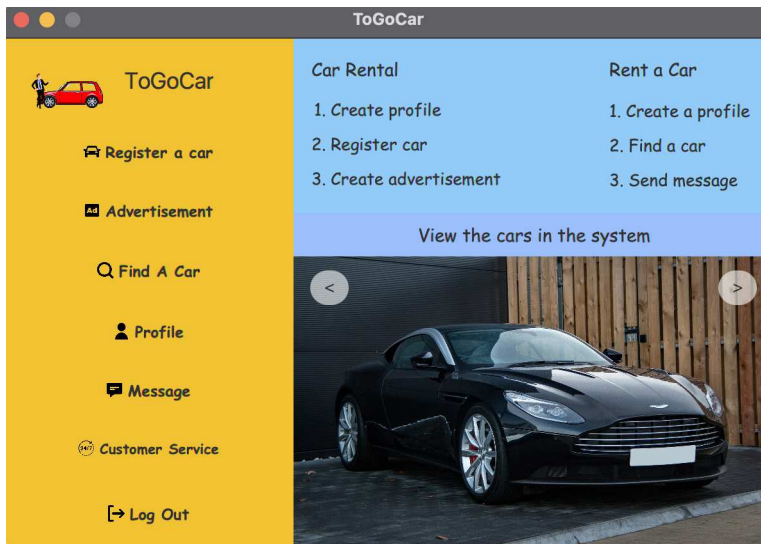
Under er en beskrivelse av hvordan man bruker applikasjonen.

Bruk av ToGoCar App:



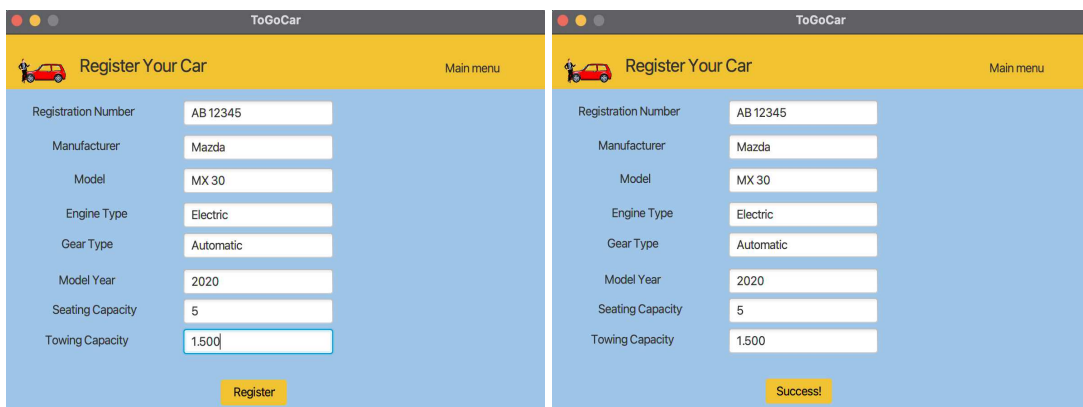
Illustrasjon: [Mann med rød bil] av Hassan, (2018).

Innloggingssiden for å få tilgang til å bruke appen. Her skriver brukeren inn e-post og passord. Hvis det ikke eksisterer noen bruker med oppgitt e-postadresse og passord, så blir de henvist videre til registreringssiden (Sign-up). Derfra blir de sendt til sin profil-side.

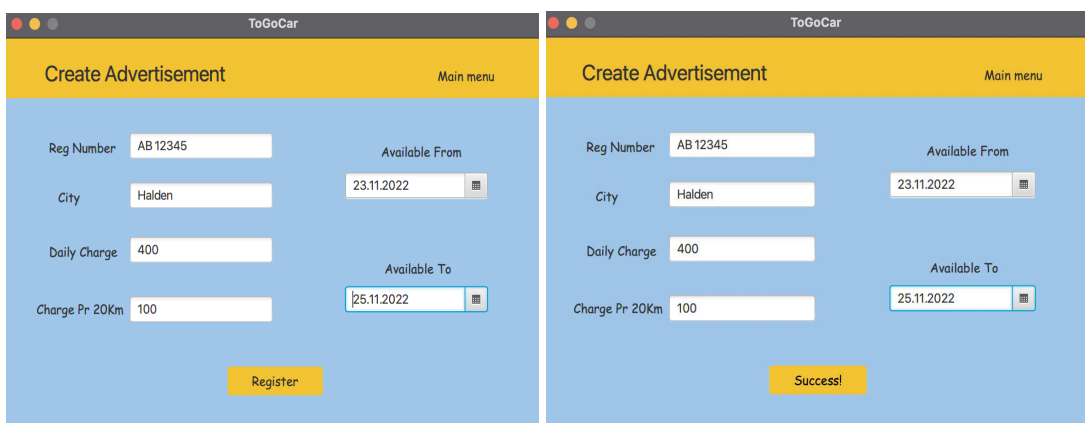


Illustrasjon: [Aston Martin Db11] av Parsons, (2017).

Hovedsiden med forskjellige funksjoner som brukeren har tilgang til.



Her vises siden for å registrere informasjon om bilen. Brukeren fyller ut all nødvendig informasjon for å registrere bilen sin i applikasjonen. Kappen “Registrer” blir endret til “Suksess”, dersom informasjonen er gyldig.



Over vises siden for å lage en bilannonse. Brukeren fyller ut informasjonen om hvor bilen befinner seg, pris per dag, og når bilen er tilgjengelig for utleie. Her vil også knappen for registrering endre seg til suksess ved gyldig oppgitt informasjon.

Dette er siden for hvordan man finner en bil. Brukeren vil skrive inn området på bilen som skal leies og velge datoene for å leieperioden. Brukeren kan også filtrere egenskaper til den tilgjengelige bilen som de ønsker å velge som f.eks. produsent, girkasse, antall seter til osv. Ved å klikke på søkeknappen vil resultatene av de tilgjengelige bilene vises, og brukeren må velge hvilken de vil leie. Det er også et alternativ for å se detaljene til bilen, se eier-profilen som leier ut bilen. Herfra kan de bestille en bil de vil ha.

<i>Owner</i>	<i>sam</i>
<i>Manufacturer</i>	<i>audi</i>
<i>Model</i>	<i>tt</i>
<i>Model Year</i>	<i>2013</i>
<i>Gear Type</i>	<i>automatic</i>
<i>Fuel Type</i>	<i>petrol</i>
<i>Seating Capacity</i>	<i>5</i>
<i>Towing Capacity</i>	<i>1500</i>

[Go Back](#)

Denne siden viser detaljer om bilen etter å ha trykket på “Se detaljer”.

Booking...

Manufacturer *audi*

Model: *tt*

Book From: *2022-11-29* **Book To:** *2022-12-02*

Total Amount: *600.0*

Choose Payment:

[Go Back](#) [Book](#)

Dette er bestillingssiden, hvor man kan velge foretrukket betalingsmåte.

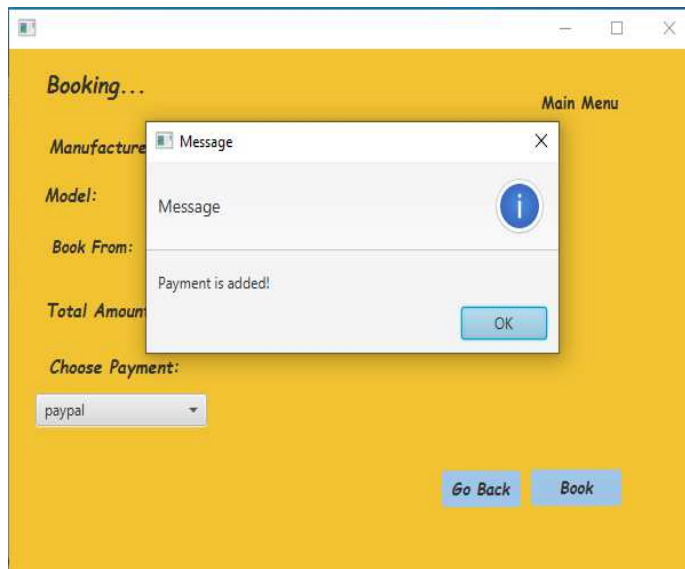
Payment Page [Return to previous page](#)

Email

Password

[Add Payment](#)

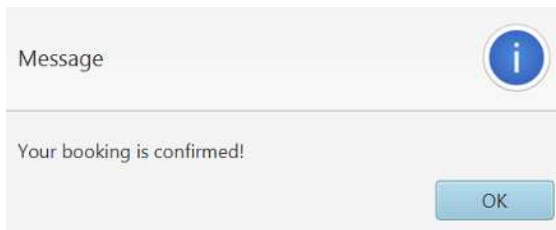
Dette er betalingssiden, der man må oppgi e-mail og passord for å legge til betalingen.



Bildet ovenfor viser bekreftelse på foretrukket betalingsmetode. Ved å velge betalingsmetode fra bestillingssiden, blir de tatt til betalingssiden. Etter å ha lagt inn informasjonen deres får de en bekreftelse på at betalingen deres er lagt til. Da kan de klikke på bestillings-knappen.

A screenshot of a "Terms Agreement" form. The form is titled "Terms Agreement" and contains the following text: "By accepting these terms, the user agrees to:", "Deposit:", "Upon booking, the deposit amount will be put on hold and charged to the rent", "Cleaning", "The vehicle will be returned as it was delivered. Failure to do so will result in a f", "Fuel:", "The vehicle will be returned with more than half a tank. Failure to do so results", "Damages:", "Damages must be reported to the owner of the vehicle and insurance compan". At the bottom of the form, there are two checkboxes: "I have read and agree to the terms" and "I do NOT agree".

Etter å ha klikket på bestillings-knappen vil brukeren bli overført til vilkårene og avtale-siden der de må godta vilkårene og betingelsene



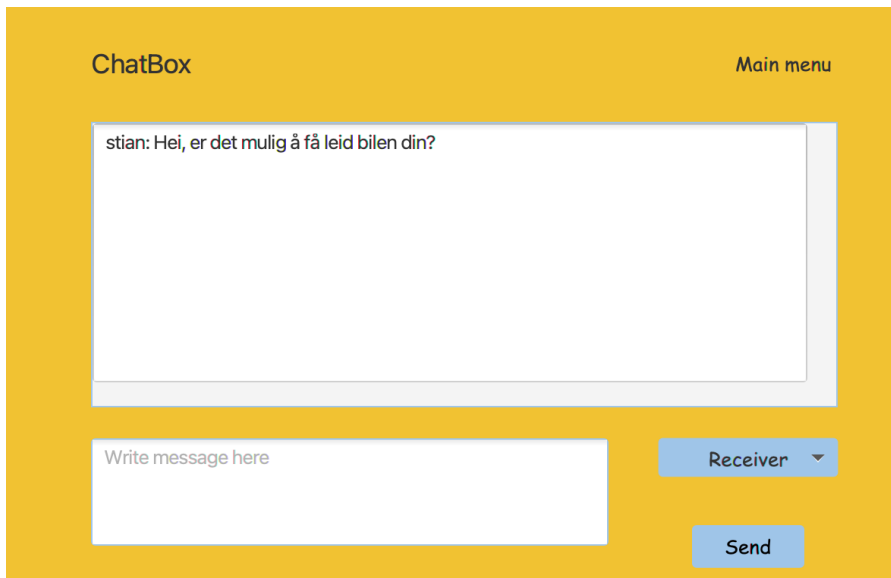
Bildet over viser en melding om at bestillingen er bekreftet etter å ha klikket på at man godtar vilkår og betingelser.

A "Profile Details" form on a yellow background. It includes input fields for First name, Last name, Email add, Password, Phone number, Bank Acnt Number, and Post number. There is also a "DL info:" section with fields for Driving licensed number, Date of Issue, and Country of Issue. An "Update Profile" button is at the bottom.

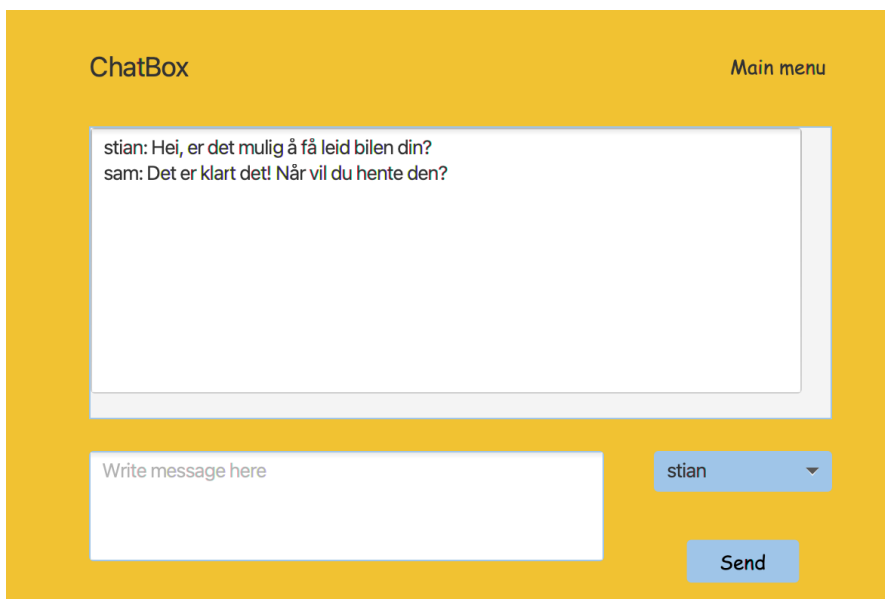
På denne siden kan brukeren redigere profilen sin hvis de ønsker å endre informasjon om profilen sin.

A "ChatBox" interface on a yellow background. It features a large text area for messages, a smaller input field with the text "Hei, er det mulig å få leid bilen din?", a dropdown menu showing "sam", and a "Send" button.

Meldingssystem hvor bruker "Stian" er innlogget og sender en melding til brukeren "Sam".

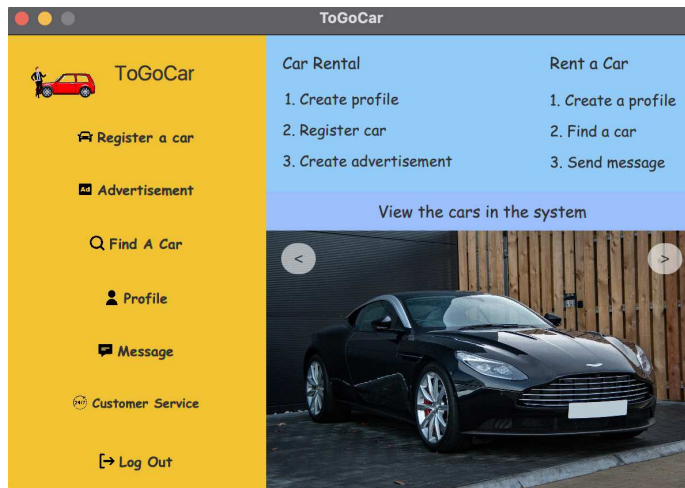


Sam logger inn via sin brukerprofil og ser at han har mottatt en melding fra Stian.



Illustrasjon etter at Sam har sendt et svar tilbake til Stian.

Systemet er ikke begrenset slik at kun to personer kan sende meldinger til hverandre. Ved å trykke på "mottaker" nedtrekksmenyen, kan man velge å sende melding til alle brukere som er registrert i systemet. På denne måten kan flere enn to personer kommunisere i en og samme meldingsboks.



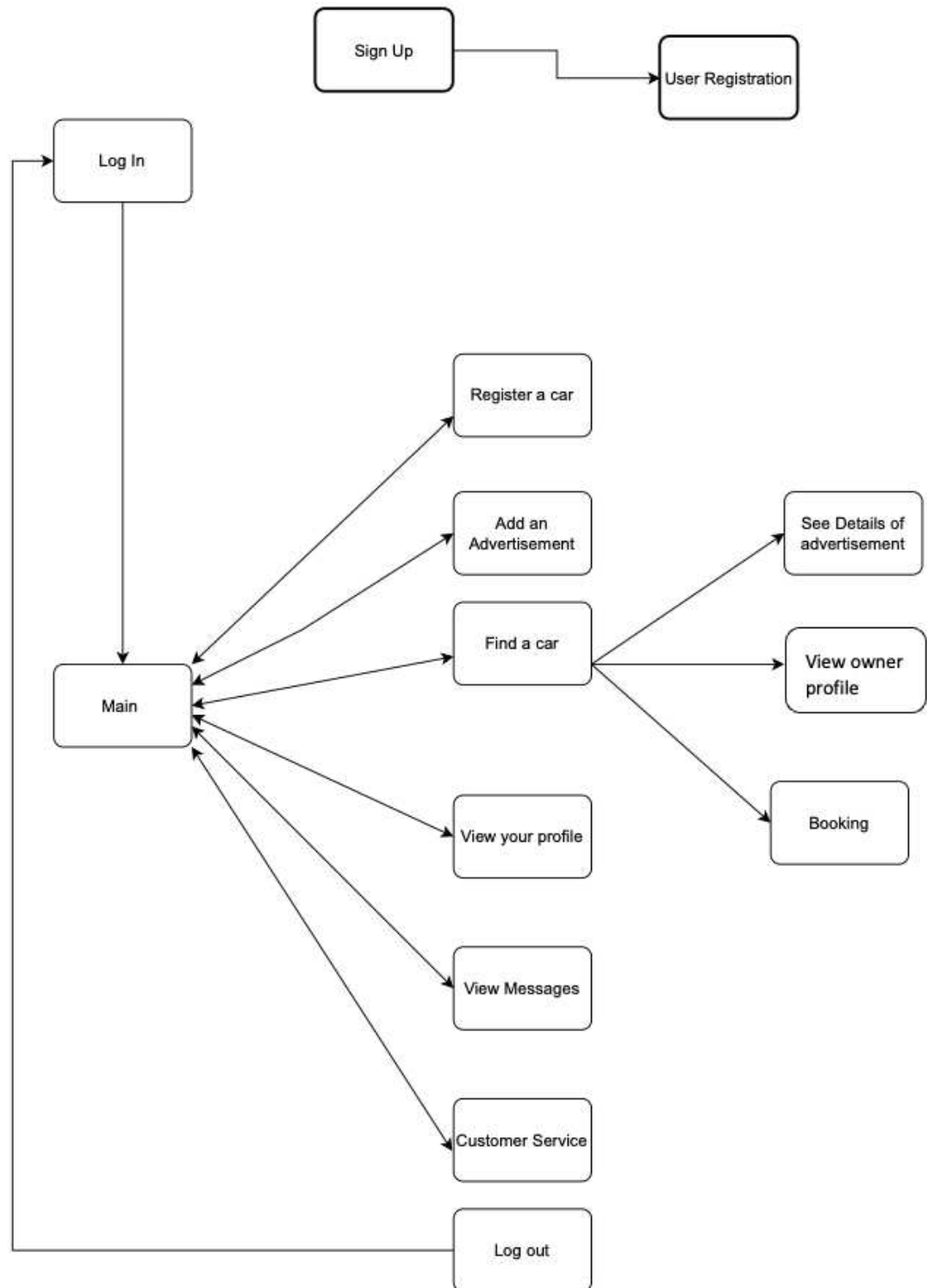
Når brukeren er ferdig med å bruke appen, kan de klikke på logg ut-knappen i venstre hjørne for å avslutte i appen.

10.2 Avgrensninger

Prototypen er ikke et fullstendig ferdig system, som vil si at det er noen funksjoner fra kravlisten som vi ikke har prioritert å implementer. De implementerte funksjonene som systemet har er basert på estimeringen vi gjorde av kravene. Funksjoner om gdpr som krav 8 og 18, samt krav 6 og 11 er ikke tatt med, ettersom at vi ikke anser de som de aller viktigste funksjonene for at systemet skal fungere og gi verdi. Krav 27 er også et krav vi ikke har prioritert, da vi har fokusert på de funksjonelle kravene en bruker kan foreta seg i systemet fremfor det visuelle.

10.3 Modellering

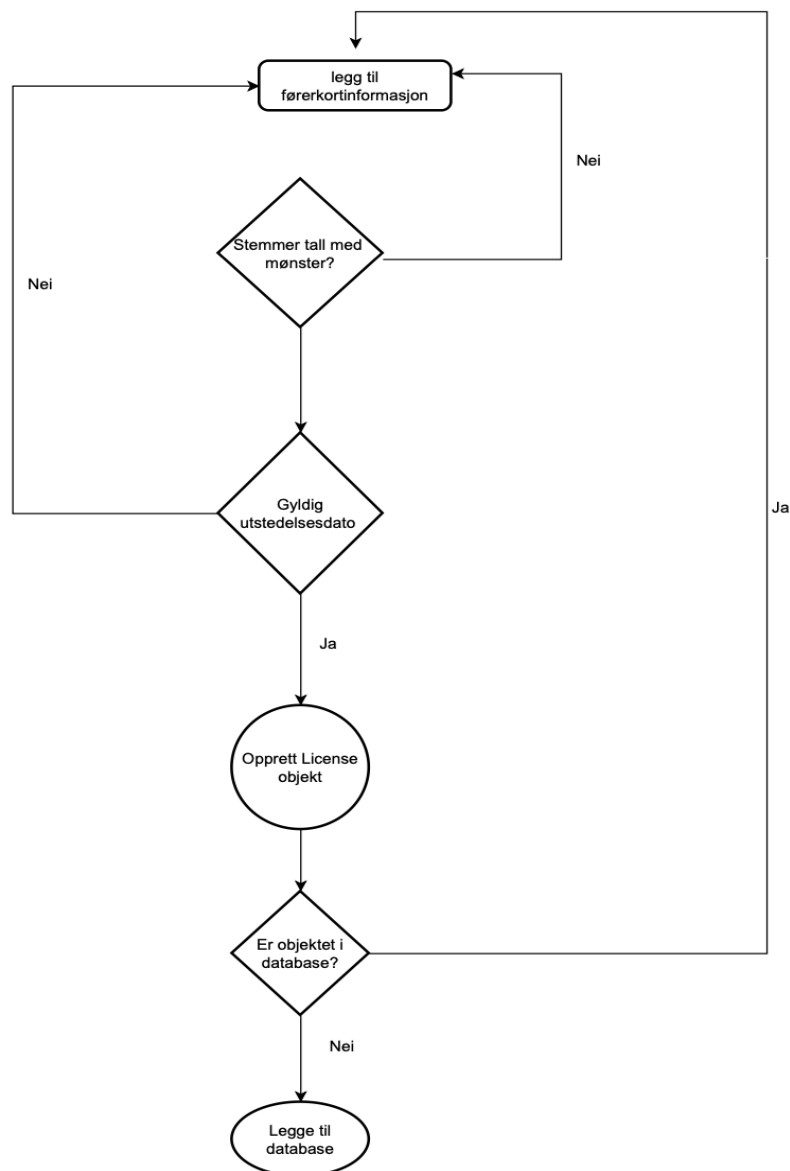
Flytdiagram: Sidene i systemet



Diagrammet viser en oversikt over de ulike sidene en bruker kan bla seg igjennom i prototypen. For å kunne bruke systemet, må man registrere seg ved å opprette en brukerkonto. Deretter kan man logge seg inn og ut av systemet etter behov. Fra hovedsiden kan man f.eks. klikke seg videre inn til siden for å registrere en bil eller til

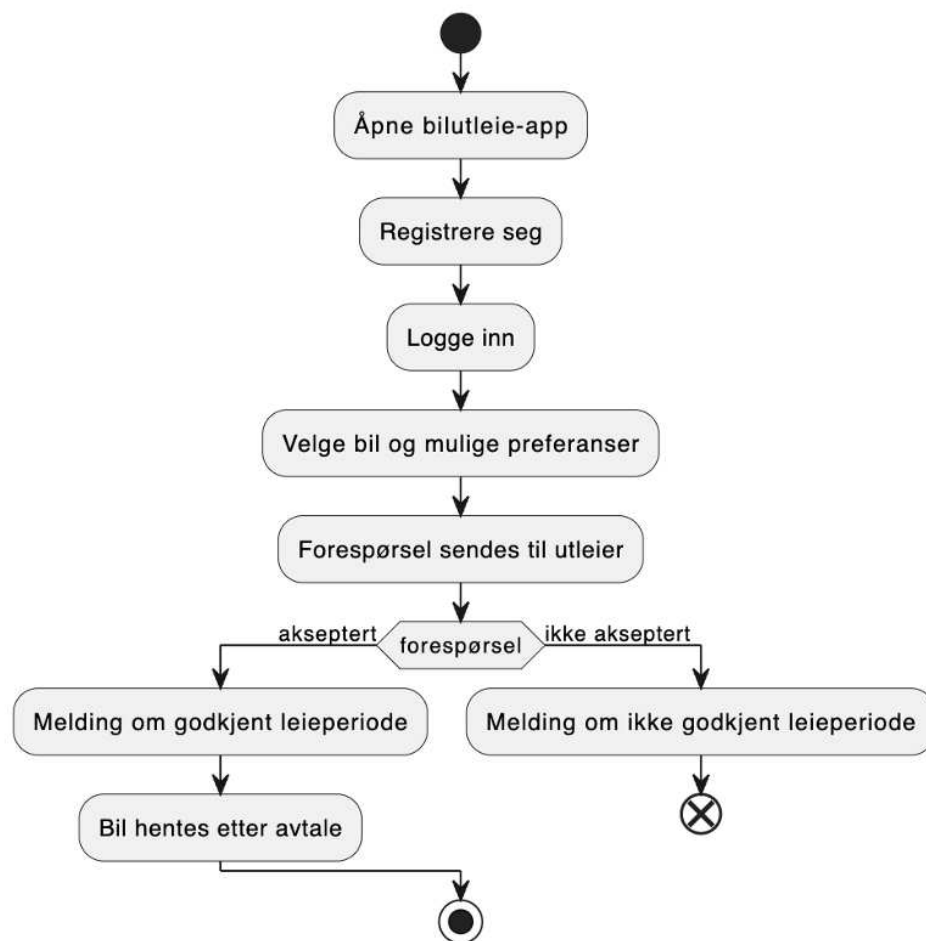
siden for å legge til en annonse. Det finnes også en egen side som viser en oversikt over brukerens profil, samt en side for å se meldinger og en for å kontakte kundeservice. I tillegg kan man fra hovedsiden gå til en ny side for å finne seg en bil å leie. Derfra kan man igjen undersøke annonsene nærmere ved å trykke seg inn på siden som viser annonsen med ytterligere detaljer rundt bilen. Det er også mulig å klikke seg inn på en side som viser bileierens profil fra den siden som viste mulige biler å leie. Fra siden som viser mulige biler kan man derfra gå til en ny side for å leie seg en bil.

Flytdiagram: Legge inn førerkort



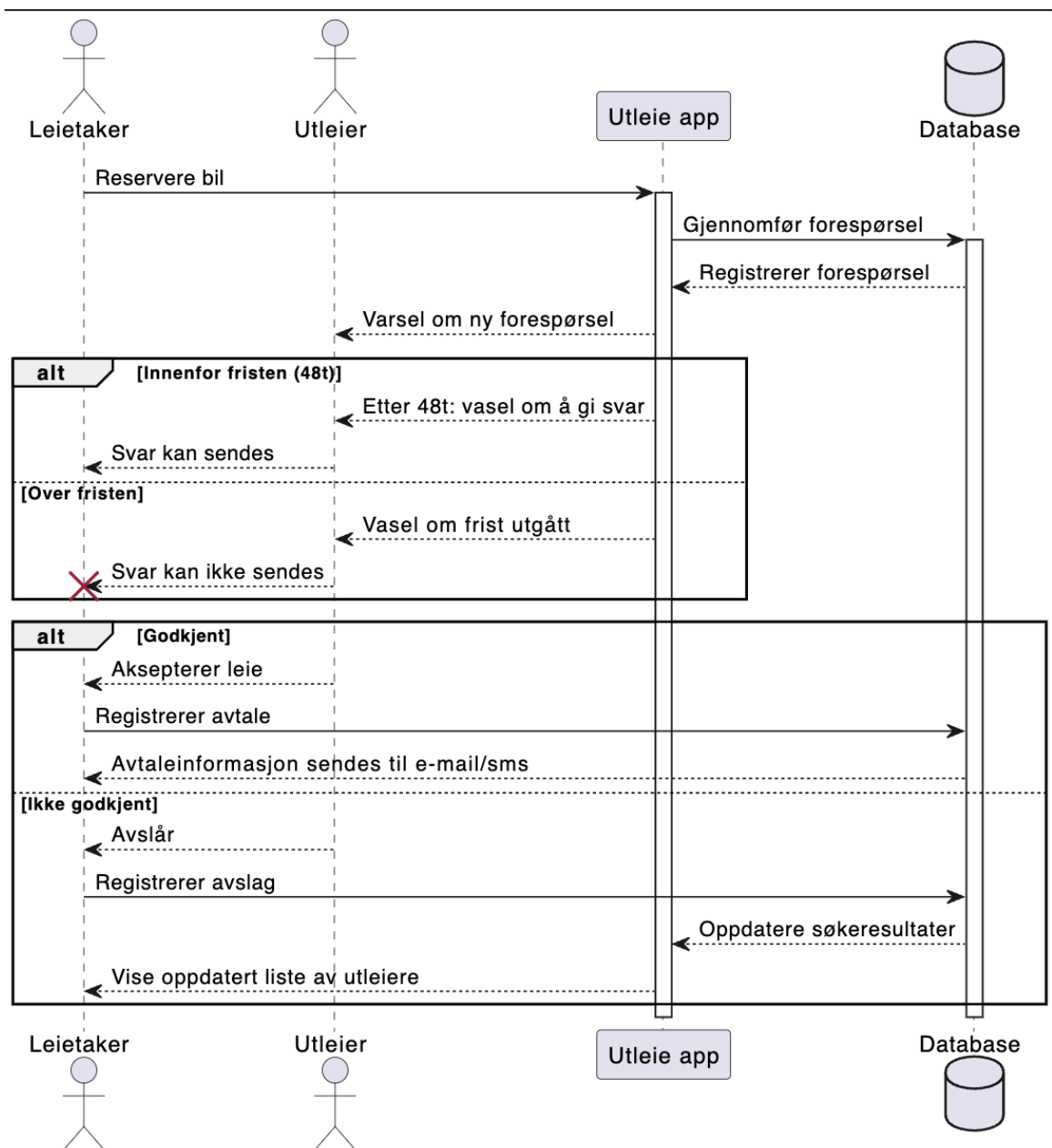
For å kunne legge til et førerkort må man oppgi nødvendig informasjon som stemmer med tallmønsteret, dersom det ikke stemmer, blir brukeren bedt om å starte prosessen på nytt. Det samme gjelder for førerkortets utstedelsesdato. Dersom alt er gyldig kan man gå videre og et lisensobjekt blir opprettet og lagret i databasen. Hvis det allerede skulle ligge inne i databasen fra før av vil brukeren bli sendt tilbake til starten av prosessen (krav 17).

Aktivitetsdiagram: Leie en bil



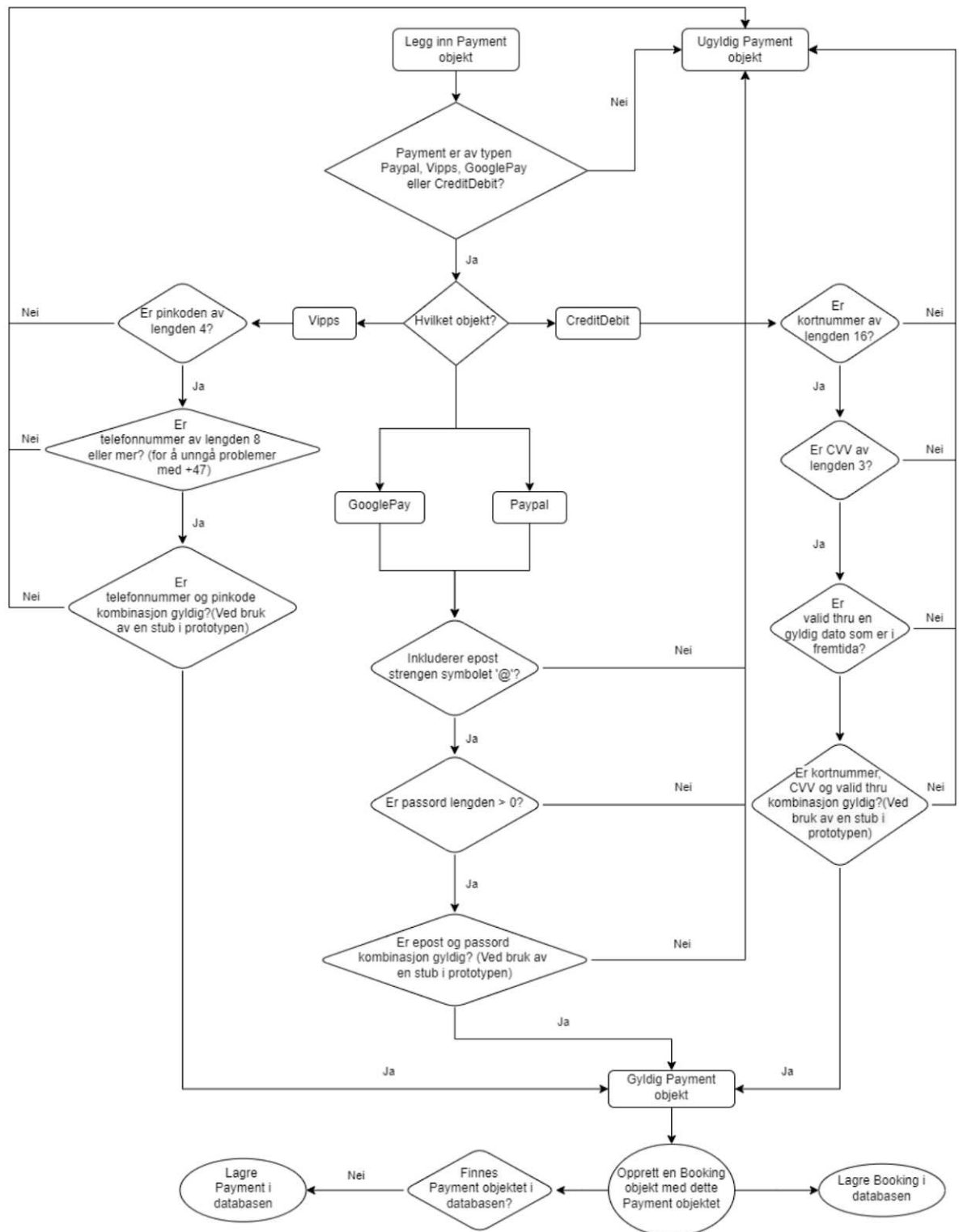
Aktivitetsdiagrammet viser en overordnet beskrivelse av systemets funksjoner som dekker krav 2, 12, 13 og 16. Den viser hvordan en leietaker kan gå frem for å leie en bil. Leietakeren må opprette en konto for å kunne leie en bil, deretter vil han kunne filtrere søkeresultatene basert på gitte preferanser. Forespørselen må godkjennes av utleier for at leietakeren skal kunne leie seg en bil.

Sekvensdiagram: Godkjenning av en forespørsel



Sekvens-diagrammet viser hva som skjer dersom en leietaker ønsker å reservere en bil, samt fristen utleier er nødt til å forholde seg til (krav 21). Diagrammet viser i tillegg alternative resultater av mulige utfall. Konsekvensen av å ikke svare leietaker innen fristen vil medføre til at utleier ikke kan sende et svar til leietaker. Dersom forespørselen blir godkjent vil det resultere i at informasjon blir sendt som varsel på e-post, fra krav nr.16. Hvis avtalen ikke blir godkjent, vil leietaker måtte starte prosessen på nytt, men få en oppdatert liste av utleiere i søkeresultatet(krav 23).

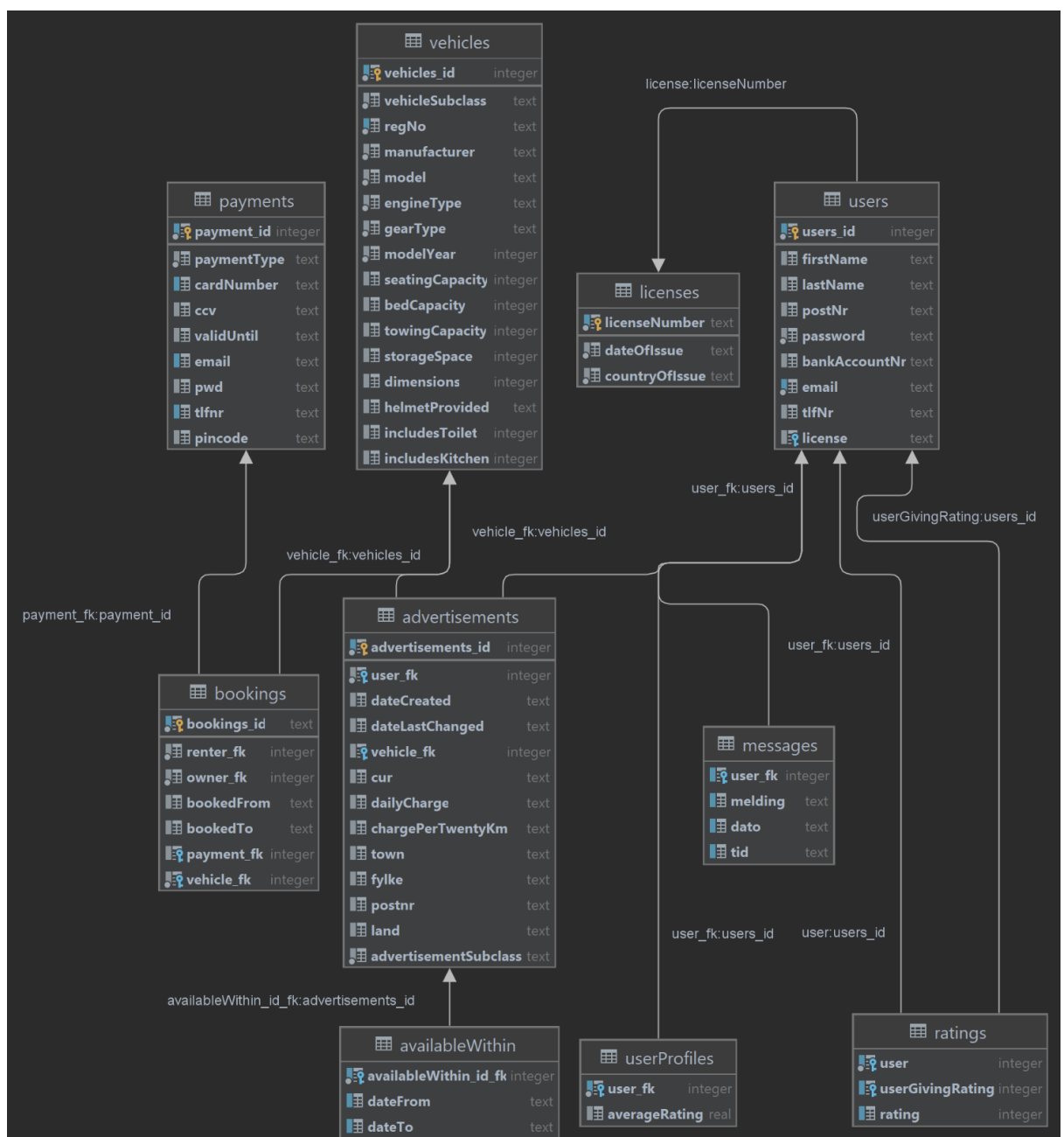
Aktivitetsdiagram: Legge inn en betaling



Diagrammet viser at for å kunne få betalt for en leid bil må en leietaker legge inn et foretrukket betalingsobjekt av enten Vipps, GooglePay, Paypal eller kredittkort (Krav

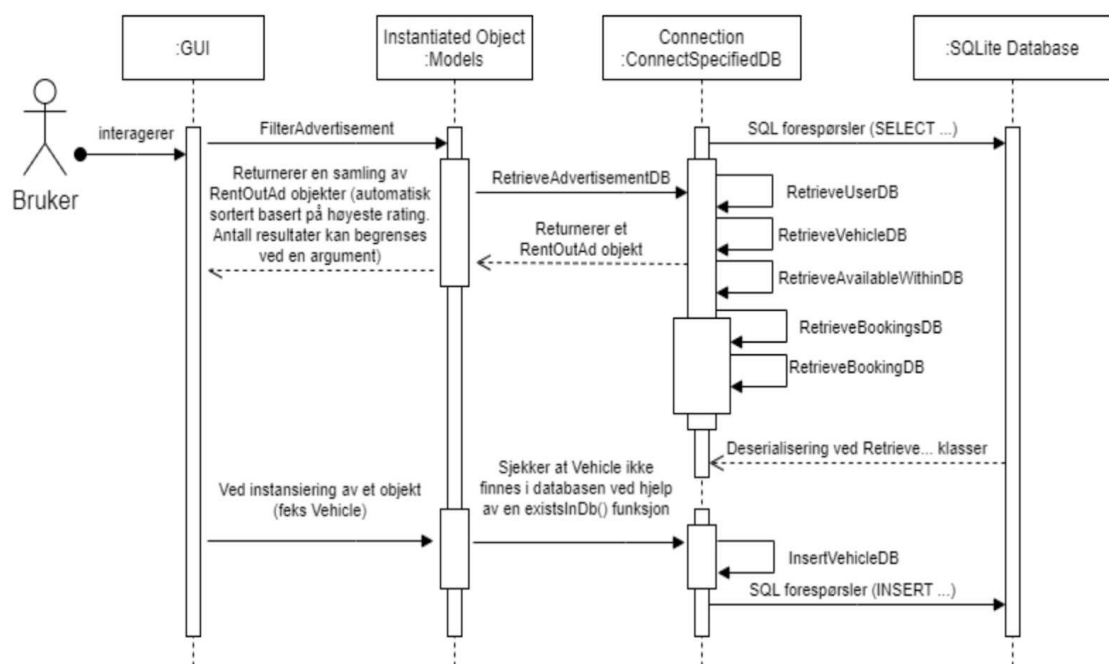
1). Avhengig av type betaling man velger, vil det føre til at man må oppgi relatert informasjon som kreves for at betalingen skal kunne gjennomføres. Diagrammet tar også hensyn til hva som skjer dersom man f.eks taster feil kode i Vipps eller om man foretar en skrivefeil, som å ikke inkludere symbolet @ ved e-post registrering i GooglePay eller Paypal. Ved feiltakelser vil man ikke kunne gjennomføre betalingen for leieperioden, siden systemet anser det som et ugyldig betalingsobjekt (Payment objekt). Dersom Payment objektet er gyldig vil et booking objekt opprettes og det lagres i databasen sammen med Payment objektet.

Database-tabeller



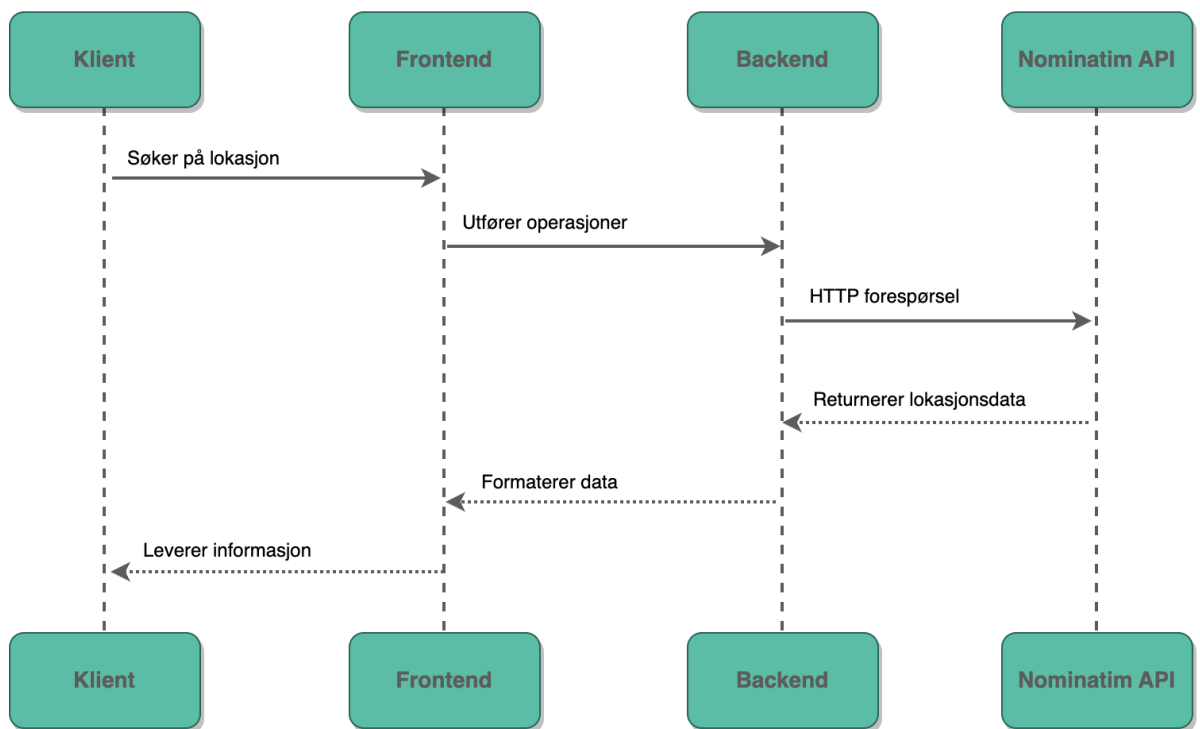
Her viser vi en samling over de ulike databasetabellene og deres relasjon til hverandre(krav 17.4). Tabellen Bookings har en kobling til databasetabellene Payments og Vehicles. Bookings henter dataen i payment og vehicle som fremmednøkler for å koble de tre tabellene sammen. Advertisements har en kobling til tabellene Vehicles og Users ved at det er laget to fremmednøkler fra disse tabellene inn i tabellen Advertisements . Samt er availableWithin relatert til advertisements. Tabellene message, userProfiles og ratings henter dataen user_id som er laget som en fremmednøkkel i tabellene fra tabellen users. Tabellen users henter data fra tabellen licenses når den skal lagre lisensen til de enkelte brukerne.

Sekvensdiagram: Interaksjon med databasen.



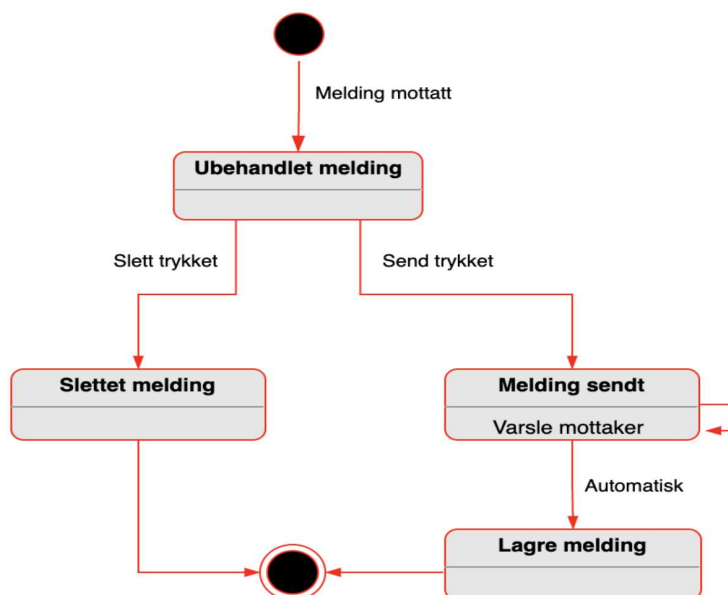
Sekvensdiagrammet viser flyten av hvordan databasen samhandler med resten av systemet når en bruker interagerer med det(17.4). Ved ulike typer handlinger fra brukeren i systemet medfører det til at SQL-forespørsler blir laget og sendt til databasen for å hente ut data som skal vises til brukeren i GUI-et. For eksempel når brukeren ønsker å hente ut annonser basert på høyest vurdering (rating).

Sekvensdiagram: Lokasjonssøk



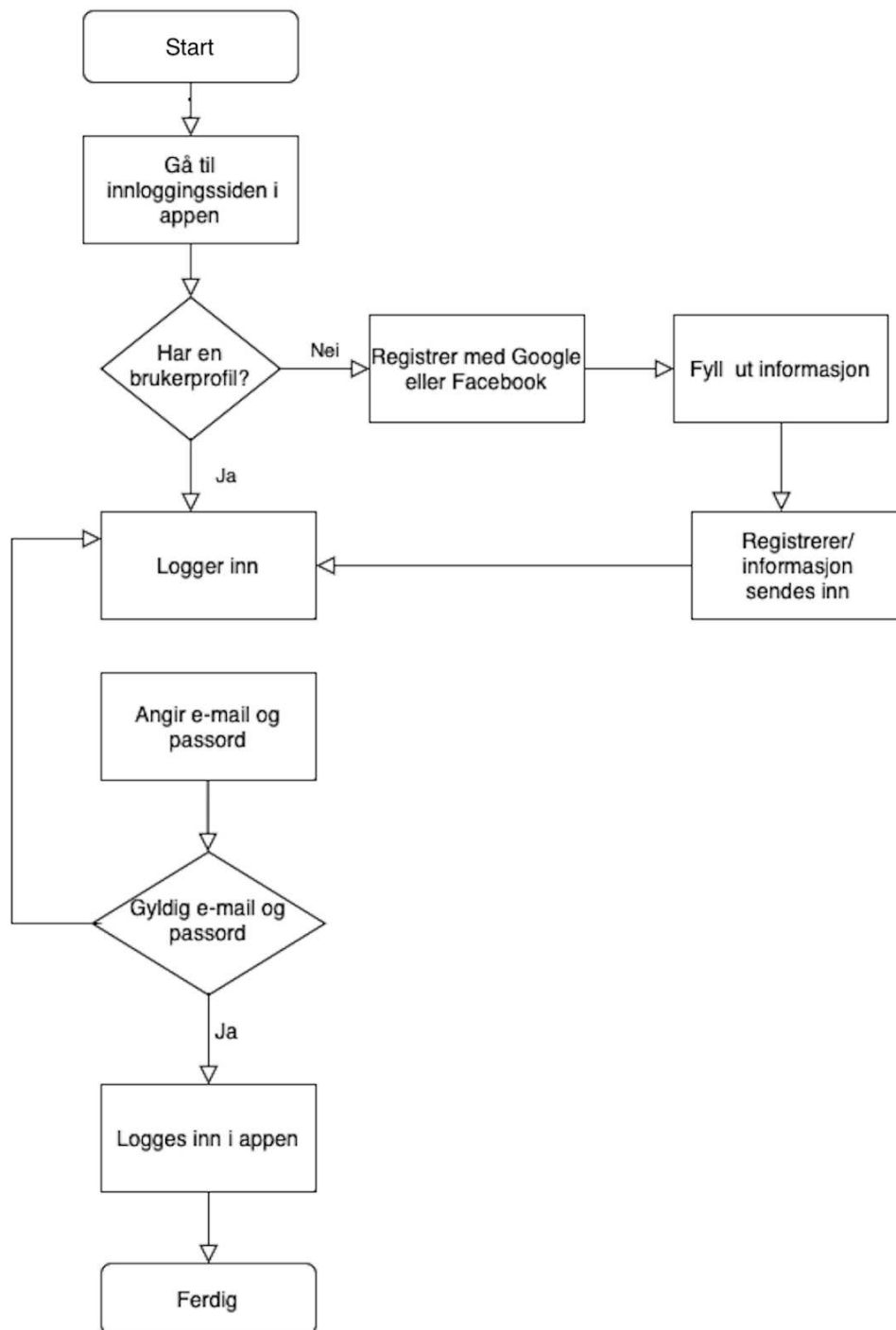
Diagrammet viser hva som skjer dersom en bruker foretar et søk på lokasjon i applikasjonen. Funksjonen henter lokasjonsdata basert på en http forespørsel til API-et. Backend formaterer dataen til frontend-delen av systemet som videre viser informasjonen som er forespurt til brukeren.

Tilstandsdiagram: Meldinger



Tilstandsdiagrammet viser hva som skjer dersom en bruker ønsker å sende en melding til en annen bruker. Den blir automatisk lagret i systemet dersom den er sendt, og man har også valget om å slette en melding (krav 12 og 17.1).

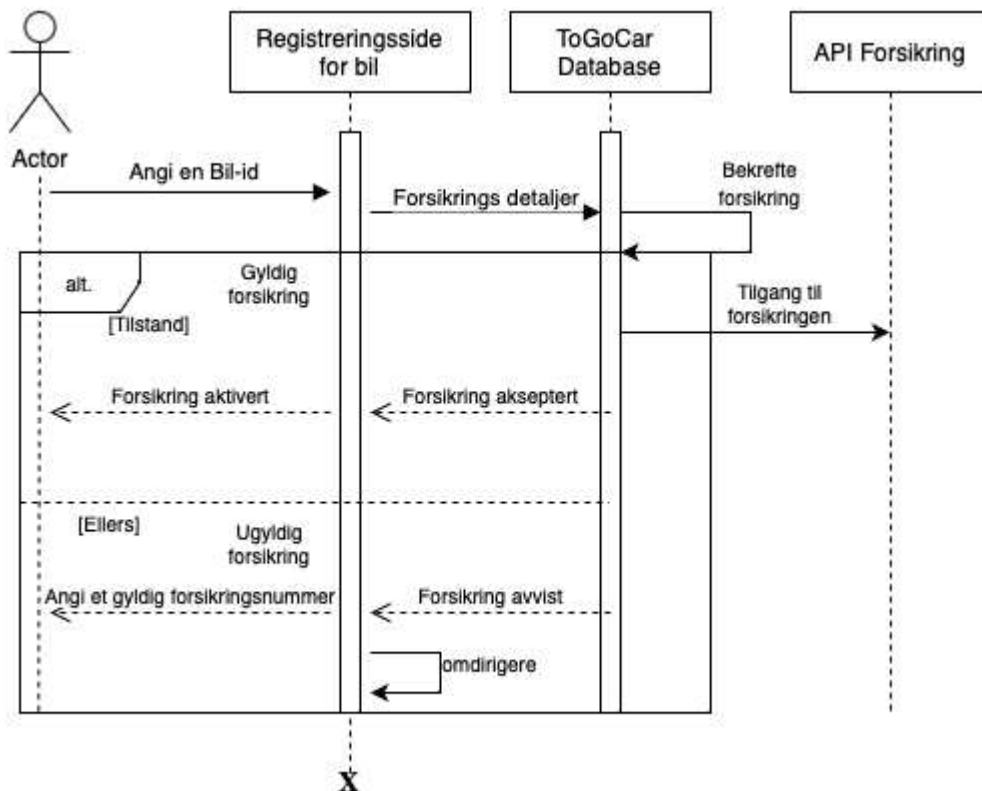
Flytdiagram: Registrering og innlogging i applikasjonen



Dette flytdiagram viser hva når brukeren skal logge seg inn i systemet.

Systemet be om kun e-post og passord, dersom brukeren allerede har registrert seg. Brukeren blir så logget inn i applikasjonen, hvis det er gyldig passord og e-mail. Har ikke brukeren registrert seg fra før, vil systemet be brukeren om å registrere seg med Google eller Facebook. Da må brukeren oppgi nødvendig opplysninger for så å bli logget inn.

Sekvensdiagram: Registrere en forsikring



Sekvensdiagrammet illustrerer hva som skjer når brukeren ønsker å registrere en forsikring. Brukeren må først oppgi en Bil-id, for så å oppgi forsikringsdetaljer som blir sendt til ToGoCar databasen. Dersom forsikrings detaljene er gyldig får API tilgang til forsikringen, og forsikringen blir akseptert og aktivert. Ved å oppgi en ugyldig forsikring vil brukeren bli bedt om å oppgi et gyldig forsikringsnummer for så å bli omdirigert til å starte prosessen på nytt.

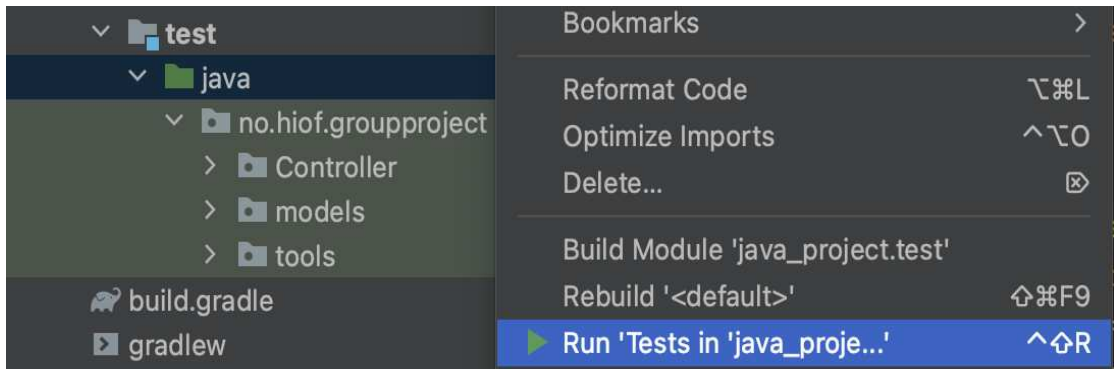
10.4 Avhengigheter

Vi har benyttet byggesystemet “Gradle”, slik at alle avhengigheter blir installert automatisk ved oppstart av prosjektet.

10.5 Tester

Demonstrasjon av hvordan testene kan kjøres i IntelliJ:

1. Åpne prosjektet i IntelliJ utviklingsmiljøet.
2. Naviger til test mappen: src/test.
3. Høyreklikk på test mappen og trykk “Run test in java_project”.



For å få testene til å kjøres fra kommandolinjen må man bruke kommandoen (se på README.md i prosjektmappa for en fullstendig guide hvordan man kjører tester i Command Prompt om man oppdager komplikasjoner):

```
gradlew test --tests "no.hiof.groupproject.*"
```

Alle 137- testene fungerer blir godkjent ved kjøring.



10.6 Svakheter

Vi har bemerket oss flere svakheter i systemet. Krav 25 om oppetid 24/7 og 26 om å støtte flere forespørsler samtidig var vanskelig å teste. For krav 26, så støtter SQLite flere brukere samtidig, men kun én bruker kan skrive til databasen for å overholde noen av ACID prinsippene. Krav nr. 13 er implementert i GUI-en, men vi har valgt å ikke differensiere kontoer på dette tidspunktet, fordi både en Google konto og Facebook konto i praksis er en gyldig e-post og passord kombinasjon.

Det er blant annet ikke mulig å laste opp bilder av leiebilen ved registrering av kjøretøyet. En bruker kan heller ikke opprette en annonse før kjøretøyet er registrert. Dette må skje etter. Vi har heller ikke fått utviklet at brukeren kan sende en

forespørsel om å leie en bil via et knappetrykk. I tillegg blir ikke brukere varslet når de mottar en ny melding. De må fysisk gå inn i meldingstjenesten og se om de har fått noen nye meldinger.

En annen svakhet er at søk på lokasjon kan gi uventede resultater da Nominatim ikke fungerer helt optimalt. Ved bruk av filtrering, må brukeren sette ønskede filtre på før søket gjøres. Filtrerings-funksjonen vil ikke fungere dersom brukeren gjør endringer i filtreringen etter at søket er gjort. Samt har vi ikke fått systemet til å spørre om autentisering når leietaker skal betale. Systemet inneholder heller ingen sikkerhetsmekanismer som en ekte applikasjon ville hatt, som for eksempel inndata validering, autentisering og kryptering av data. Vi har fått laget en test på krav 5, men funksjonen blir en svakhet da det ikke er implementert i GUI-et til prototypen.

Som tidligere nevnt under 7. Estimering er gdpr nedprioritert i utviklingen av systemet. Vi ser på det som en svakhet at dette ikke er med fra start, da det er noe enhver bedrift bør tenke på og ha med i sine systemer. Funksjon nummer 8 «Leietaker og utleier (brukere av systemet) skal kunne etterspørre GDPR data fra administrator/firmaet av systemet.». Her kan dataene som blir lagret om brukeren enkelt hentes ut av utvikleren av systemet med en spørring, men det er ikke laget en funksjon i GUI-et som gjør at brukeren kan foreta en forespørsel selv fra systemet. Det samme gjelder for krav 18 «Systemet skal følge GDPR». Det er viktig å ta hensyn til at brukerne ønsker å vite hva av data som blir fanget opp av systemet og formålet med det, ettersom at de har rett til det ut ifra personopplysningsloven. Det er en lov alle virksomheter bør ta stilling til ved oppsamling av sensitive persondata, siden man kan bli hardt straffet dersom man ikke følger loven.

11. Kildeliste

Prototypen:

Abiyyu A, H. (u.å.). *24 Hours Support free icon*. Flaticon.

https://www.flaticon.com/free-icon/24-hours-support_5167002?term=247&page=1&position=3&page=1&position=3&related_id=5167002&origin=search

Apien. (u.å.). *Comment free icon*. Flaticon.

https://www.flaticon.com/free-icon/comment_3193061?term=message&page=1&position=40&page=1&position=40&related_id=3193061&origin=search

Freepik. (u.å. a). *Car free icon*. Flaticon.

https://www.flaticon.com/free-icon/car_89102?term=car&page=1&position=5&page=1&position=5&related_id=89102&origin=search

Freepik. (u.å. b). *User free icon*. Flaticon.

[https://www.flaticon.com/free-icon/user_456212?term=user&page=1&position=1&position=1&related_id=456212&origin=search](https://www.flaticon.com/free-icon/user_456212?term=user&page=1&position=1&page=1&position=1&related_id=456212&origin=search)

Hassan, M. (2018). [Mann med rød bil]. Pixabay.

<https://pixabay.com/no/illustrations/bil-salg-bilforretning-bilanlegg-3189771/>

Icon Small. (u.å.). *Advertisement free icon*. Flaticon.

https://www.flaticon.com/free-icon/advertisement_8283208?related_id=8283208&origin=search

Lakonicon. (u.å.). *Log out free icon*. Flaticon.

https://www.flaticon.com/free-icon/log-out_3808346?term=log%20out&page=1&position=57&page=1&position=57&related_id=3808346&origin=search

Muhammad, F. (2018). [Audi A6]. Pixabay.

<https://pixabay.com/no/photos/audi-a6-sedan-fire-d%C3%B8rs-bil-3234422/>

Parsons, T. (2017). [Aston Martin Db11]. Pixabay.

<https://pixabay.com/no/photos/aston-martin-db11-bil-aston-martin-2977862/>

Those Icons. (u.å.). *Magnifying Glass free icon*. Flaticon.

https://www.flaticon.com/free-icon/magnifying-glass_2089805?related_id=2089805&origin=search

torquedetail. (2021). [BMW X5]. Pixabay.

<https://pixabay.com/no/photos/svart-bil-bil-bmw-bmw-x5-6097934/>

Tumisu. (2018). [Smilende forretningsmann]. Pixabay.

<https://pixabay.com/no/photos/tommelen-opp-lykkelig-mann-smilende-3640323/>

Dokumentasjon:

Uutilsynet. (u.å.). *WCAG 2.0-standard*. Tilsynet for universell utforming av ikt.

Hentet fra

<https://www.uutilsynet.no/wcag-standard/wcag-20-standard/86>