# Prototype

Protoypen vårt har blitt laget ved bruk av Java forms. Vi lagde den på plattformen IntelliJ IDEA. Java Forms gir brukeren en ganske lett GUI som man kan sette opp sitt program i. Hvis man vil jobbe videre med protoypen, eller vil bare lese kildekoden så kan man lett gjøre det ved bruk av IntelliJ. For å utføre testing så brukte vi «dependenciene»; JUnit Jupiter API (scope: test, version 5.7.2), JUnit Jupiter engine (scope: test, version 5.7.2), JUnit Jupiter params (scope: test, version 5.7.2) og til slutt maven-failsafe-plugin (versjon 2.22.2). Vi brukte også en siste dependancy kalt gson. Denne lot oss serialisere og deserialisere objekter til JSON så vi kunne skrive/lese til/fra fil. IntelliJ kan bli nedlastet her: <https://www.jetbrains.com/idea/download>.

## Veiledning for å åpne prosjektet (direkte metode)

1. Trykk «Open» på IntelliJ idea:



1. Finn frem prosjektmappen (kalt SEG):

Et bilde som inneholder tekst

Automatisk generert beskrivelse

1. Så lett som det så skal det være mulig å se på kildekoden/jobbe videre med protoypen:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

## Veiledning for å sette opp Maven (i tilfelle 7.1 ikke fungerer)

1. Trykk «New Project» på IntelliJ.

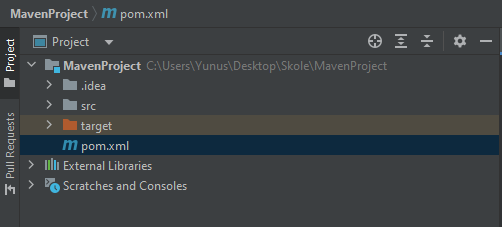


1. Her så skal du velge egendefinert navn og lokasjon. Deretter velger du Java som språk, og Maven som build system. Behold versjon 17 på openjdk. Trykk create når ferdig.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

1. Nå skal du over til pom.xml som du kan finne på listen til siden.

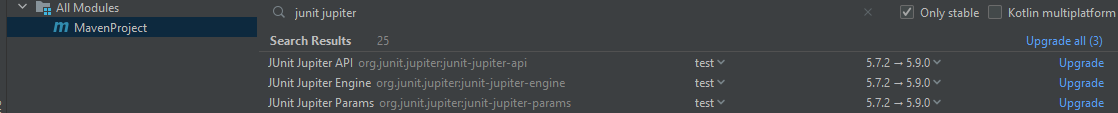


1. Du kan enten kopiere koden under og build prosjektet, eller manuelt legge til dependencies. Koden ligger på side 23. Ved å dobbel trykke «bildet» så får du muligheten til å kopiere (usikker på hvordan funksjonaliteten fungerer på pdf). Trykk dependencies som finnes nederst i vinduet. Her søker du på «junit jupiter».

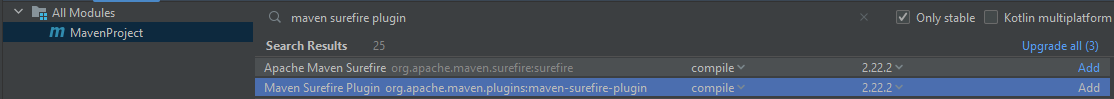
Et bilde som inneholder tekst, elektronikk, datamaskin, skjermbilde

Automatisk generert beskrivelse

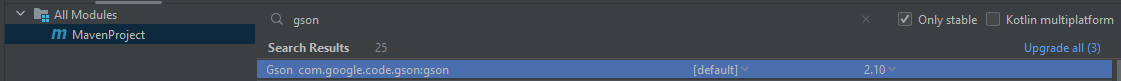
1. Bytt fra compile til test på; engine, api og params. Du skal i tillegg sette alle 3 versjonene til 5.7.2. Deretter kan du trykke add på hver av dem:



1. Nå søker du etter maven surefire plugin og legger til den:



1. Til slutt så søker du etter gson og legger til den:



1. Nå er alt klart. Hvis gson for en eller annen grunn ikke fungerer (slik som det gjorde med oss på starten), så kan du fikse det ved å gå over til «maven» helt til høyre i vinduet og trykke på «reload» knappen:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

### Litt om prototypen

Protoypekoden er delt opp i 2 hovedmapper for kildekoden, og 2 egne javafiler for testing. Det var tidligere en egen mappe for Interfaces, som vi prøvde å sette opp funksjoner som vi kunne kalle til for å skrive til fil/lese fra fil. Dette fungerte dessverre ikke slik som vi ville, og har derfor blitt fjernet i sluttversjonen av protoypen. Endringene hvor vi brukte FileHandler og slikt kan finnes i git endringene. Vi fikk til eventuelt til å skrive til fil, men dessverre så fikk vi ikke til å lese fra fil. Dette kan finnes i versjon 3.4

Et bilde som inneholder tekst

Automatisk generert beskrivelse

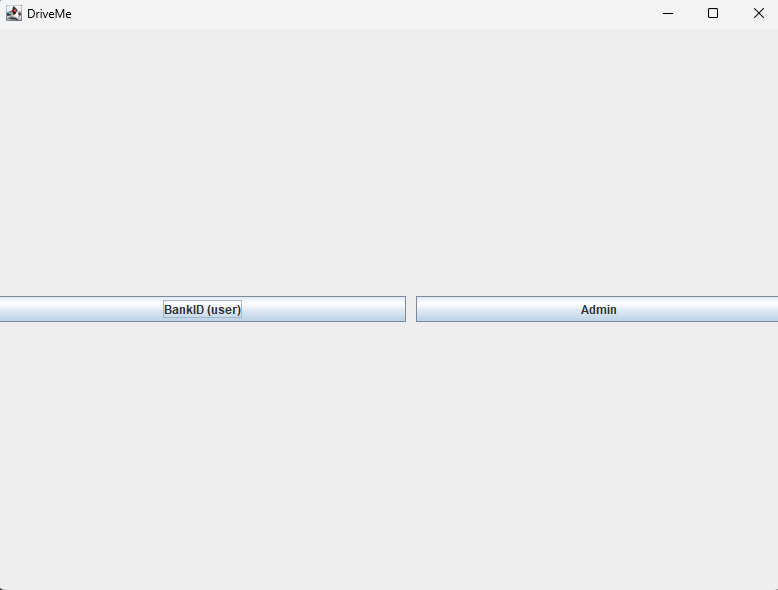
I forms mappen så kan vi finne både forms filen som ble brukt til å designe GUI-en. I tilegg så er javafilen som kontroller forms i samme sted. Denne javafilen kontrollerer alt fra hva som blir vist, når det blir vist og hva som skjer når vi trykker på knapper og samhandler med protoypen.

I modules mappen så kan vi se begge klassene som har blitt brukt; CarRegistration og CarRepository. Først så lages et CarRepository som blir brukt til å lagre bilene som blir laget ved bruk av CarRegistration.

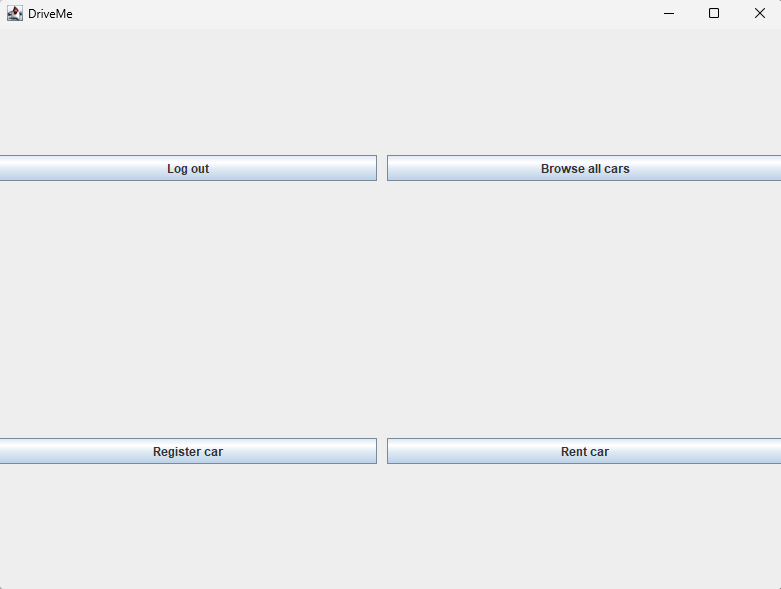


## Prototypens layout og funksjon

Når du først åpner prototypen så blir du velkommen av logg inn siden. Her har du muligheten til å logge inn som bruker (BankID), som gir deg muligheten til å registrer og låne bil, og admin, som gir deg muligheten til å redigere og slette biler fra listen.



Hvis du logger inn som bruker så får du muligheten til 4 funksjoner. Første er å logge ut (som tar deg tilbake til forrige siden), andre er å se alle biler som er registrert, tredje er å registrere en bil mens den siste er å kunne bestille en bil.



Vi starter først med «Browse all cars». Her så kan du se alle biler. Både de som er «rented» og de som ikke er «rented». Ved å trykke på main menu så går du alltid tilbake til innloggingssiden til brukeren.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

I register car panelet så har man 5 forskjellige valg. Ut ifra de 5, så har man et begrenset antall valg på 3 av dem. Disse 3 inkluderer antall «seats» man kan velge, hva slags «transmission type» det er på bilen, og hva slags «engine type» det er på bilen. Disse har allerede blitt fast bestemt av oss på hva vi aksepterer. Vi har gjort det umulig å kunne registrere biler uten registrerings nummer, men utenom det så er alt mulig. Man kan også velge om bilen er klar til å bli lånt ut med en gang, eller om du allerede har lånt den ut.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Et bilde som inneholder tekst

Automatisk generert beskrivelseEt bilde som inneholder tekst

Automatisk generert beskrivelseEt bilde som inneholder tekst

Automatisk generert beskrivelse

Til slutt så har vi «rent car» panelet. Her får du en liste med alle biler som er tilgjengelige. Du kan nå velge hvilken bil fra listen du vil låne bare ved å trykke på bilen, og deretter velge betalingsmåte.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Til slutt så har vi «admin» panelet. Når du først går inn i panelet så får du listen over alle biler i listen, både tilgjengelig og utilgjengelig. Her kan du lett trykke på en bil og slette den fra listen.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Du har også muligheten til å redigere biler ved å trykke på en bil og trykke «edit car». Her har du muligheten til å bytte alt informasjon slik som du ønsker ettersom at du er admin.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

# Testing ved bruk av Junit

**Tittel**:

Hvis du registrerer en bil uten å røre «rented» sjekkboksene så skal bilen automatisk være «not rented».

**Antagelser:**

Brukeren glemte å trykke på «not rented».

**Test steg:**

1. Bruker legger til bil uten å trykke på sjekkboksene.

**Forventet resultat:**

Bilen blir automatisk lagret som not rented.

**Junit test:**



**Tittel**:

Når du låner en bil så skal det bli faktisk bli lånt ut.

**Antagelser:**

Brukeren låner bilen.

**Test steg:**

1. Bruker låner bilen.

**Forventet resultat:**

Bilen blir lånt ut og kommer nå ikke opp på «availableCarsList».

**Junit test:**



**Tittel**:

Når du registrer en bil som du vil legge til «repository» så blir den faktisk lagt til «repository»-en.

**Antagelser:**

Brukeren registrerte bil.

**Test steg:**

1. Bruker registrerer bil.

**Forventet resultat:**

Bilen kommer inn i listen for bil oppbevaringsstedet.

**Junit test:**



**Tittel**:

Hvis admin fjerner bil fra «repository»-en så skal den bli fjernet.

**Antagelser:**

Admin vil fjerne en bil

**Test steg:**

1. Admin sletter bil fra «repository».

**Forventet resultat:**

Bilen blir slettet og befinner seg ikke lenger i «repository»

**Junit test:**



**Tittel**:

Sjekker om «repository» skrives til fil og leses opp. Sjekkes ved å se om «repository» er fortsatt samme klasse etter å ha blitt lest opp igjen

**Antagelser:**

Om fil skrivning og fil lesing fungerer riktig.

**Test steg:**

1. Blir gjort via backend.

**Forventet resultat:**

Listen blir skrevet ned og lest opp riktig, og sjekkes om listen er tom, og om listen er samme klasse som før den ble skrevet ned.

**Junit test:**



**Tittel**:

Bruker skal ikke kunne legge til 2 biler med samme registreringsnummer

**Antagelser:**

Bruker prøver å legge til 2 biler med samme registreringsnummer

**Test steg:**

1. Bruker lager 1 bil med registreringsnummer PG46886
2. Bruker prøver å registrere en bil til med registreringsnummer PG46886

**Forventet resultat:**

Bilen blir ikke lagt til og vi sjekker ved å se om bilen eksiterer i listen.

**Junit test:**



**Tittel**:

Bruker skal bare få opp tilgjengelige biler i listen når bruker skal låne bil

**Antagelser:**

Bruker vil låne ledig bil

**Test steg:**

1. Bruker åpner panelet for å låne bil og ser en ledig liste.

**Forventet resultat:**

Alle bilene som kommer opp er bare ledige biler. Sjekkes ved å som opptatte biler kommer frem i «getAllAvaialableCars» listen.

**Junit test:**



**Tittel**:

Alle biler, både tilgjengelig og utilgjengelig, blir vist når det trengs.

**Antagelser:**

Bruker/admin vil se alle bilene som har blitt registrert

**Test steg:**

1. Bruker/admin åpner listen for alle biler

**Forventet resultat:**

Alle biler, både tilgjengelig og utilgjengelig, blir vist.

**Junit test:**

