**Prosjektinnlevering – Gruppe 9**

Emil Berglund

Andreas B. Olaussen

Khalid H. Osman

Sebastian W. Thomsen

Ida K. Tollaksen

Høgskolen i Østfold

Ord: TBD

Emnekode: ITF20319

Prototypedokumentasjon – Andreas   
Brukerhistorier og Personas - Andreas

[Introduksjon: Emil 4](#_Toc182916871)

[Problemstilling: Emil 5](#_Toc182916872)

[Løsningsbeskrivelse: Emil 6](#_Toc182916873)

[Kjernefunksjonalitet: 6](#_Toc182916874)

[Åpen og fleksibel platform 6](#_Toc182916875)

[Data og tilgang 6](#_Toc182916876)

[Ekspanderbarhet og antakelser 6](#_Toc182916877)

[Begrensninger 7](#_Toc182916878)

[Funksjonelle krav: 9](#_Toc182916879)

[Ikke-funksjonelle krav: 10](#_Toc182916880)

[Brukere av systemet 11](#_Toc182916881)

[Systemarkitektur 13](#_Toc182916882)

[Prototypedokumentasjon: 14](#_Toc182916883)

[Testing: 15](#_Toc182916884)

[Utviklingsprosess (må se an denne) 16](#_Toc182916885)

[Instruksjoner for bruk: 17](#_Toc182916886)

[Konklusjon: 18](#_Toc182916887)

[- Modeller og vedlegg: 19](#_Toc182916888)

[Hva skal leveres: 20](#_Toc182916889)

Innholdsfortegnelse

Sjekkliste – Husk!

* Dokumentasjon skal være lesbar og forståelig for ALLE, også de uten kunnskap innenfor programmering eller IT. Definer derfor begreper og forklar nøye.
* Dokumentasjon er viktig. Sleng på så mye kommentarer som dere klarer i koden deres!
* Sørg for at det er beskrevet hvordan en person uten dyptgående IT-kunnskap kan bygge, kjøre og teste prototypen deres.
  + Avhengigheter i applikasjonen deres skal kunne installeres automatisk ved hjelp av et pakkesystem for språket eller rammeverket dere benytter (f.eks. maven eller gradle om dere bruker Java, pip eller poetry  
    for Python)
* Hver deltaker i gruppa skal også lage et individuelt vedlegg (som et separat dokument) som inneholder en beskrivelse av den valgte måten å organisere gruppa på, arbeidsfordeling og av hvem som har gjort hva. Fokus skal ligge på hvordan organisasjonen av arbeidet i gruppa fungerte, både positive sider, negative sider, arbeidsfordeling og lærdom til senere prosjekter. Det skal også inkluderes en timeliste for hele prosjektet. Disse vedleggene skal inkluderes samlet i den felles gruppeinnleveringen.
* Husk at dere skriver et løsningsforslag for oppdragsgiver, slik at ting kan bli utviklet og de/dere som skal utvikle forstår hva som skal gjøres - det er ikke en rapport om hva dere har gjort (utover kapittelet om prototypen til slutt og de individuelle vedleggene).
* Det skal ikke tekniske forklaringer inn i prosjektdokumentasjonen. Hvis dere begynner å prate om ORM-er, biblioteker, etc. og implementasjon så er det sannsynligvis feil plass.

# Introduksjon: Emil

I dagens samfunn ser vi en stadig økende interesse for smarte løsninger som gjør hverdagen enklere og mer effektiv. For oppstartsbedrifter som ønsker å tilby innovative produkter, kan det imidlertid være utfordrende å utvikle løsninger som både er brukervennlige og tilpasset målgruppens behov, samtidig som løsningen differensierer seg selv på markedet fra andre aktører.

Denne dokumentasjonen tar utgangspunkt i en oppstartsbedrift som ønsket å utvikle et smart system for strømvarsling og optimal lading av elbiler. Målet er å lage en løsning som er enkel å bruke for folk med begrenset IT-kompetanse, samtidig som den leverer avanserte funksjoner som strømprisanalyse, varslinger, og anbefalinger om når det lønner seg å lade bilen.

Dokumentet beskriver hele utviklingsprosessen fra kravspesifikasjon til ferdig prototype. Videre beskriver det kjernesystemet og funksjonene som er nødvendig for at kunden skal forstå hvordan produktet fungerer, og hvordan man kan bygge videre på det. Det inneholder detaljer om systemets funksjonalitet, tekniske krav, arkitektur, testing og hvordan vi har gått fram for å komme fram til en løsning som tilfredsstiller kundens behov.

Denne dokumentasjonen har derfor to formål: Å demonstrere hvordan et teknologisk konsept kan gå fra idé til prototype, og å vise hvordan et solid fundament kan legges for videreutvikling. Det er vår forhåpning at dette arbeidet ikke bare dekker oppdragsgivers behov, men også inspirerer til videre innovasjon innen smart energistyring og bærekraftig teknologi.

# Problemstilling: Emil

Problemstillingen vi sto ovenfor som ledet oss til idéen vår handlet mye om alminnelighet og åpenhet. I dag er det ikke uvanlig at strømaktører har egne applikasjoner og tjenester for sine strømkunder. I disse applikasjonene finner man som regel funksjoner som smart-lading og andre muligheter for assistanse rundt lading. Disse er derimot veldig lukkede løsninger og krever ofte ekstra tilbehør eller fysiske produkter fra aktøren direkte, eller fra produsenter som støttes av strømselskapet. Ordene «lukkede løsninger» refererer her til at funksjonene som man noen ganger finner i slike applikasjoner, ikke er tilgjengelig for alle.

Vi tenkte derfor: «Hvordan kan vi utvikle en åpen og brukervennlig løsning for elbillading som eliminerer avhengigheten av proprietære løsninger og spesifikke strømselskaper, samtidig som den tilbyr avanserte funksjoner som optimalisert lading og strømprisanalyse?»

# Løsningsbeskrivelse: Emil

For å møte utfordringen i dagens marked med lukkede løsninger for elbillading, er det valgt å tilrettelegge for en åpen og brukervennlig tjeneste som er uavhengig av spesifikke strømselskaper eller eksternt tilbehør. Hovedmålet er å tilby en fleksibel løsning som tilpasser seg brukerens behov, uten tekniske barrierer, samtidig som den gjør lading enklere.

## Kjernefunksjonalitet:

Produktets kjernefunksjonalitet er å varsle brukeren når bilen bør lades. Dette kan gjøres på flere måter, men hovedsakelig basert på disse verdiene:

* Tidligere, gjeldene og fremtidige strømpriser
* Prognoser for prisutvikling
* Gjenværende batteriprosent på bilen

## Åpen og fleksibel platform

I motsetning til eksisterende løsninger som Vibb og Tibber, skal ikke dette systemet kreve noen tilknytning til en spesifikk strømleverandør eller innkjøp av kompatible produkter. Dette gjør løsningen tilgjengelig for alle, uavhengig av hvilket system eller leverandør de bruker.

## Data og tilgang

For at tjenesten skal fungere optimalt kreves det visse data og tilganger fra brukeren. Disse er hovedpunktene:

* Posisjonstilgang: For å avgjøre når brukeren ankommer hjemmet.
* Strømdata: For nåværende og estimerte priser.
* Batteriprosent: For å sikre at varsler er relevante og ikke sendes unødvendig.

## Ekspanderbarhet og antakelser

Produktet bør kunne utvikles videre basert på brukerbehov og tilbakemeldinger. Det er her viktig å ikke overkomplisere løsningen under utviklingen. Eksempler på fremtidige utvidelser kan derimot være som følger:

* Smart varsling: Tilpassede varsler, basert på kjørevaner og batterinivå.
* Fysisk produkt: En skjerm i garasjen, som aktiveres når bilen parkeres som presenterer brukeren med sanntidsdata, og gir anbefaling.

## Begrensninger

Ettersom løsningens hovedmål er enkelhet og åpenhet, vil det være klare og tydelige begrensninger for den. Dette er noen typiske begrensinger som denne løsningen kan støte på:

1. Begrenset tilgang til data.
   1. Strømprisdata: Nøyaktige og oppdaterte strømpriser kan være vanskelige å få tak i, dersom strømleverandøren ikke deler denne informasjonen via åpne API-er. Dette kan føre til at man må nøye seg med generaliserte priser fra åpne API-er, som ikke vil være tilpasset brukerens valg av strømselskap.
   2. Batteriprosent: For enkelte bilmodeller er det nødvendig med spesiell programvare eller tillatelse fra bilprodusent for å hente ut batteridata, noe som kan begrense den universelle kompatibiliteten.
2. Avhengig av brukerens tillatelser
   1. Løsningen krever tilgang til brukerens posisjon og bilens data. Mange brukere kan være skeptiske til å gi slike tillatelser på grunn av personvern og sikkerhetsbekymringer.
3. Begrenset teknologi hos brukeren
   1. Ikke alle brukere har smarttelefoner eller apper som er kompatible med moderne API-er og systemer. Brukere med eldre biler eller telefoner uten smart-funksjonalitet kan også møte utfordringer med å bruke løsningen.
4. Kompatibilitet og standarder
   1. Elektriske biler og ladeløsninger varierer mye mellom produsenter og leverandører. Hvis det ikke finnes en standardisert metode for datainnhenting og kommunikasjon, kan løsningen ha begrenset bruk for noen biltyper eller ladestasjoner.
5. Pålitelige prognoser for strømpris
   1. Fremtidige strømpriser er ofte uforutsigbare og kan påvirkes av faktorer som vær, etterspørsel, og energimarkedet. Feilaktige prognoser kan føre til at brukeren mottar dårlige anbefalinger, noe som svekker tilliten til løsningen.
6. Irriterende varsler
   1. Selv med tilpasning kan varsler oppleves som irriterende hvis de er for hyppige eller ikke relevante (f.eks. hvis brukeren bare har kjørt en kort tur). Dette kan føre til at brukeren skrur av funksjonaliteten.
7. Juridiske og regulatoriske barrierer
   1. Tilgang til strømdata og bilinformasjon kan være underlagt nasjonale lover og regler, som kan variere fra land til land. Dette kan hindre en global lansering av løsningen uten tilpasninger.

# Funksjonelle krav:

## Identifisering av nødvendige funksjoner

Funksjonelle krav beskriver hvilke funksjoner systemet krever for å oppfylle både brukerens behov og løse oppdragsgivers problemstilling. Tabellen viser en oversikt over kravene, med status for om funksjoner er implementert

|  |  |  |
| --- | --- | --- |
| **ID** | **Beskrivelse** | **Status i prototypen** |
| FK1 | Bruker skal kunne registrere seg med både e – post og passord | Implementert |
| FK2 | Bruker skal kunne logge inn med e – post og passord | Implementert |
| **FK3** | **Systemet skal kunne hente strømpriser fra et åpent API** | Implementert |
| FK4 | Bruker skal motta varslinger når strømpriser er lav nok til optimal ladning | Implementert |
| FK5 | Systemet skal vise nåværende, fremtidige og historiske strømpriser i en graf | Implementert |
| FK6 | Systemet skal kunne identifisere brukerens posisjon og registrere når bruker har ankommet hjem | Implementert |
| FK7 | Bruker skal kunne gi tillatelser for sin posisjon og varsler ved første gangs bruk | Implementert |
| FK8 | Bruker skal kunne både legge til og se informasjon om registrerte biler i et dashbord | Delvis Implementert |
| FK9 | Bruker skal kunne se anbefalinger om når det lønner seg for bruker å lade basert på både strømpris og batteri | Delvis Implementert |

## Funksjoner relatert til brukeren

1. **Registrering og innlogging (FK1, FK2)**

Brukeren skal kunne opprette konto med e-post og passord. Bruker skal også få kunne logge inn for å få tilgang til personlig tilpasset strømdata og varsler. Dette skal videre føre til en sikker tilgang til brukerens data. **Implementasjon:** Firebase Authentication håndterer autentisering. Funksjonene er testet og sikre

Det fungerer slik at når bruker registrer seg, lagres e-post og passord i Firebase Authentication. Dette gir en sikker og skalerbar løsning for brukerhåndtering.

Når bruker logger inn sjekker Firebase om e – post og passord stemmer overens med eksisterende konto

1. **Varslingssystem (FK4, FK7)**

Varslingssystemet har i oppgave å sikre at bruker mottar varsler når det er økonomisk nyttig å lade bilen. Dette er for å redusere strømkostnader og optimalisere ladeplan. **Implementasjon:** Firebase Cloud Messaging brukes for push varsler. Data fra Strømpris - API og posisjonskontroll trigger varsler

Det blir sendt push – varsler til brukerens mobil, ved hjelp av Firebase Cloud Messaging. Dette trigges av strømprisdata og brukerens posisjon. For eksempel: dersom prisen på strøm faller under et gitt nivå og brukeren er innenfor definerte hjem posisjonen, sendes det varsel til brukeren.

1. **Strømprishåndterings (FK3, FK5)**

Denne delen av systemet henter strømpriser og visualiserer disse i en graf. Dette er for å gi bruker en oversikt over nåværende historisk og fremtidige priser. **Implementasjon:** API – data hentes med pris tjenester klasser og diagram klassen brukes til å vise det grafiske.

Det fungerer slik at API – intergrasjon henter priser i sanntid. Deretter lagres prisene midlertidig i en database og visualiseres i diagram. **For eksempel:** kan bruker planlegge lading ved å følge med på grafen som viser hvordan strømprisen utvikler seg over tid.

1. **Posisjonsbaserte tjenester (FK6)**

GPS delen av systemet går ut på at applikasjonen sjekker om bruker er innenfor en radius definert av bruker som kalles “hjem”. Når bruker er innenfor radiusen sender den ut varsler til bruker. Dette gjørs for å unngå irrelevante varsler når bruker ikke er i den definerte “hjemme” posisjonen. **Implementasjon:** En utviklet klasse “HomeChecker” bruker Haversine – formelen for å bygge avstander.

GPS funksjonen fungerer slik at “HomeChecker” klassen sjekker avstanden mellom brukers nåværende posisjon og de definerte hjemme – koordinatene. Dersom avstanden fra brukerens posisjon er innenfor en radius på 100 meter, anses brukeren som “hjemme”. **Eksempel på scenario**: bruker ankommer hjemmet. Systemet registrer posisjonen og sender ut et varsel om ladning basert på strømpris.

1. Dashbord og ladeanbefalinger (FK8, FK9)

Dashboard delen i applikasjonen gir brukeren oversikt over brukerens registrerte biler og deres ladestatus. Brukeren får anbefalinger på lading basert på strømpriser og bilens batteristatus. **Implementasjon:** Dashbord er implementert i “DashboardPage.vue”. Anbefalinger på lading er delvis ferdig og må videre utvikles.

Denne delen av applikasjonen fungerer slik at bruker legger til biler manuelt i applikasjonens dashbord og kan se detaljer om batteristatus og forbrukshistorikk. Videre beregnes ladeanbefalinger ut ifra strømprisen og bilens gjenværende batteri nivå. **Eksempel fra applikasjonens prototype:**

Dashbord: “Volkswagen ID.3 - Batterinivå 67%”

Anbefaling: “Det lønner seg å lade senere i kveld.”

## Gruppekrav basert på funksjonelle områder

Brukerhåndtering:

* Funksjoner: FK1, FK2
* Status: Fullstendig implementert

Datahåndtering:

* Funksjoner: FK3, FK5
* Status: Fullstendig implementert

Varsling og posisjonssystem:

* Funksjoner: FK4, FK6, FK7
* Status: Fullstendig implementert

Dashboard og biler:

* Funksjoner: FK8, FK9
* Status: Delvis implementert

## Forklaring av krav som dekkes i prototypen

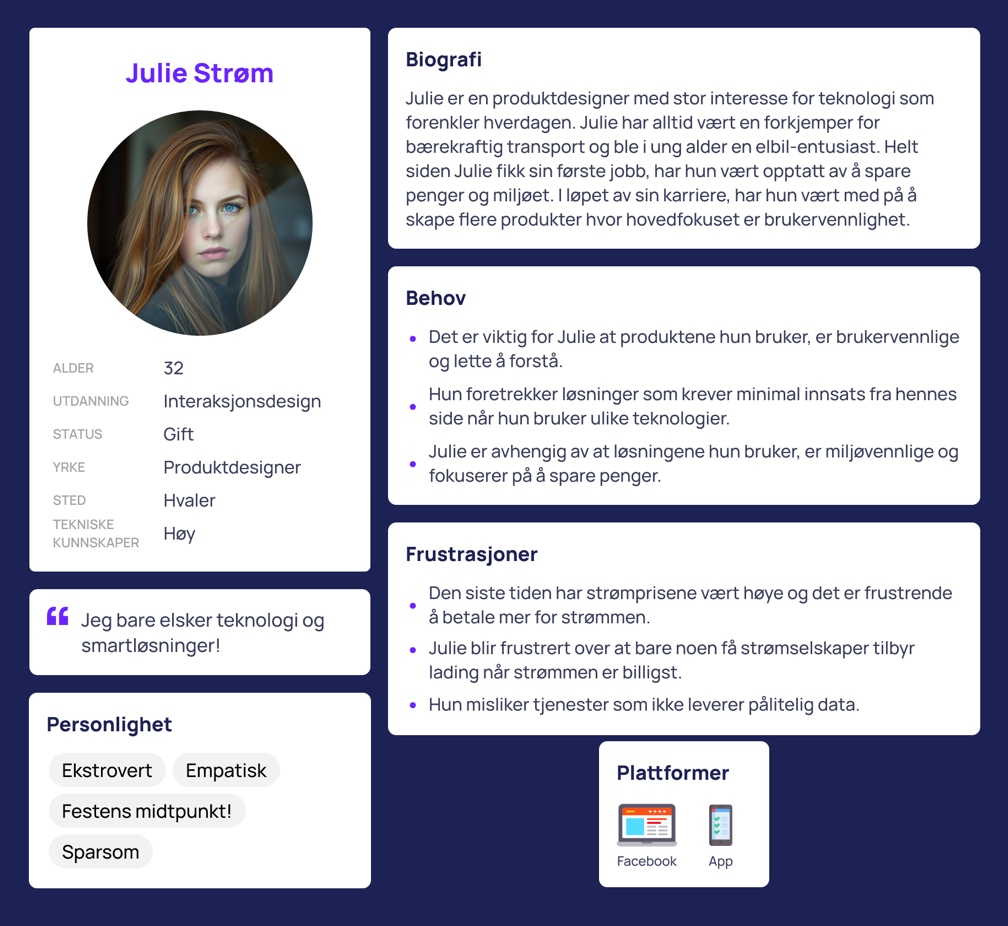
# Ikke-funksjonelle krav:

* Ytelse, sikkerhet, tilgjengelighet, og andre tekniske krav.
* Krav som ikke estimeres i nytte/kompleksitet.

# Brukere av systemet

Brukerne av vårt system er hovedsakelig el-bileiere som er opptatt av å spare både miljø og penger gjennom effektive løsninger. De er opptatt av smartteknologi og bærekraft, men har ikke nødvendigvis dyptgående it-kunnskaper. Disse brukerne prioriterer enkle og intuitive løsninger som er visuelt attraktiv og lett å forstå. Produktet vårt skal passe “mannen i gata” og være et verktøy som ikke krever opplæring og som gir god verdi til kundene.

Under er en visuell representasjon av “Julie Strøm” og “Markus Johansen” som er fremtidige brukere av systemet vi ønsker å tilby.



**Julie Strøm**

**Brukerhistorie 1: Lading når strømmen er billigst**

Som en miljøinterressert elbileier ønsker jeg å kunne planlegge lading av bilen min når strømprisen er lavest, slik at jeg kan spare penger og miljøet.

**Scenario 1: Planlegging av lading**

Julie kommer hjem fra jobb og har koblet bilen sin til hjemmeladeren sin. Hun henter seg en kopp kaffe, setter seg ned i sofaen og åpner VarsEL-appen for å sjekke dagens strømpriser. Julie ser at strømprisen er lavest mellom kl.17.00-22.00. Julie konfigurerer appen til å starte og stoppe ladingen automatisk i dette tidsrommet, slik at hun kan fokusere på andre ting resten av dagen.

**Brukerhistorie 2: Oversiktlig informasjon**

Som elbileier ønsker jeg å kunne se og legge til informasjon om bilen min i et oversiktlig dashbord, slik at jeg kan ha kontroll over lading og informasjon knyttet til bilen min.

**Scenario 2 : Registrere bilen**

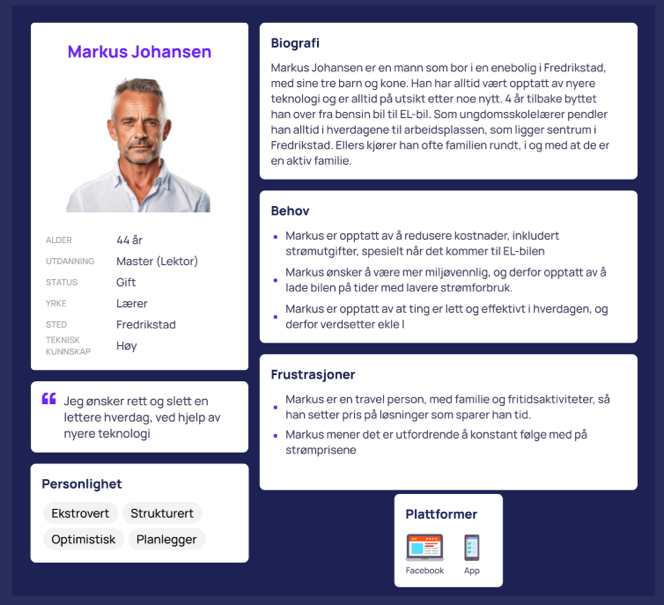
Julie har akkurat kjøpt seg ny elbil og klør i fingrene etter å registre den i VarsEL-appen sin. Hun åpner appen, navigerer seg til dashbordet, trykker på “Se all kjøretøy” og legger til den nye bilen. På Dashbordet har hun nå en oversikt over bilen sin, batteristatus, høyest og lavest strømpris, samt nåværende strømpris og mulighet til å navigere seg til en strømprisgraf.

**Brukerhistorie 3: Varsling**

Jeg ønsker å ha muligheten til å kunne bli varslet når strømprisen er lavest når jeg ikke har mulighet til å analysere strømprisene selv.

**Scenario 3: På farten**

Julie har en travel dag og er i full gang med å forberede middag til gjester og deres barn. Hun har ikke tid til å sjekke dagens strømpriser manuelt. Midt i forberedelsene får hun et varsel fra VarsEL-appen om at strømprisen er på sitt laveste. Siden bilen allerede er koblet til hjemmeladeren, spør appen om hun ønsker å starte ladingen. Med et raskt trykk på “Ja” i appen aktiverer hun ladingen og kan fortsette med middagsforberedelsene uten å bekymre seg.



**Markus Johansen**

**Brukerhistorie 1: Registrering i appen**

Som en ny bruker av VarsEL-appen ønsker jeg å registrere meg enkelt og gi tillatelser til posisjon og varsler, slik at jeg kan bruke systemet effektivt og motta varslinger om strømpriser.

**Scenario 1: Førstegangsregistrering**

Markus åpner VarsEL-appen for første gang og blir møtt av en velkomstmelding. Han trykker på knappen "Kom i gang" og blir guidet gjennom en kort registreringsprosess. Først velger Markus sin strømleverandør fra en liste. Deretter blir han spurt om å tillate varslinger om strømpriser, som han enkelt huker av. Appen ber ham deretter om å gi tilgang til posisjonstjenester, slik at funksjonene kan tilpasses der han befinner seg.

Til slutt blir Markus gitt to alternativer for registrering: å opprette en konto med e-post og passord, eller å logge inn med Google-kontoen sin. Han velger å bruke Google-kontoen for en rask registrering. Etter noen sekunder kommer han til dashbordet, der han ser en oversikt over strømpriser og får videre tilgang til appens funksjoner.

**Brukerhistorie 2: Se strømpriser i graf**

Som elbileier ønsker jeg å kunne se sanntidsstrømpriser, fremtidige- og historiske strømpriser via en graf, slik at det blir enklere for meg å se når jeg burde lade bilen.

**Scenario 2: Bruke strømgrafen**

Markus har lunsjpause på jobben og tar frem VarsEL-appen. Han navigerer seg via dashbordet til strømprisgrafen og analyserer gårsdagens, dagens og morgendagens strømpriser og gjør seg opp en mening på når han burde lade bilen sin.

**Brukerhistorie 3: Posisjonsbasert varsling**

Som bruker av tjenesten ønsker jeg å at appen automatisk registrerer når jeg er hjemme, slik at jeg får personlige varslinger om lading basert på min posisjon.

**Scenario 3: Posisjonsvarsling**

Markus kommer hjem fra kjøretur og VarsEL-appen registrerer at han er i nærheten av hjemmet sitt. Han får et varsel fra appen sin om at strømprisen er lav og om han ønsker å lade bilen. Markus trykker på “Ja” og fortsetter med det han skal gjøre resten av dagen.

# Systemarkitektur

# Prototypedokumentasjon:

* Hva prototypen inkluderer (MVP-funksjonalitet).
* Hvordan prototypen implementerer kravene (henvis til krav-ID-er).
* Forklaring av arkitekturvalg i prototypen.
* Kjente begrensninger og svakheter.

Introduksjon til prototypen:

Formål med prototypen

Omfang

Beskrivelse av funksjonalitet:

Hvilke funksjoner er implementert?   
Hvordan fungerer funksjonene?

Eksempler på brukerinteraksjon

Brukergrensesnitt og design (UI)

Visuell fremstilling

Hvordan er UI laget?

Interaktive elementer

Teknologi og verktøy brukt

Hvilke er brukt og hvordan er de implementert?

Kjente svakheter i prototypen

Hva mangler i prototypen?

Planen videre?

Testing:

* Beskrivelse av testplanen: hva som testes og hvordan.
* Eksempler på automatiserte tester som dekker funksjonelle krav.
* Hvordan tester viser feilsituasjoner og korrekt oppførsel.

# Utviklingsprosess (må se an denne)

* Hvordan teamet har jobbet (smidig metodikk og sprintplaner).
* Bruk av Kanban eller lignende for oppgavesporing.
* Oversikt over arbeidsfordeling og timebruk.

# Instruksjoner for bruk:

## Front-end (Emil) (nivå 2)

-

## Back-end (nivå 2)

### Electricity (nivå 3)

### GPS (nivå 3)

### Notifications (nivå 3)

### Users (nivå 3)

* Hvordan kjøre prototypen på en ny maskin (bygging, kjøring, testing).
* Eventuelle forutsetninger for å kjøre systemet.
* Hvordan avhengigheter installeres.

# Konklusjon:

* Modeller og vedlegg:
* Legg ved modeller her, og ellers rundt om i teksten, hvor det er relevant og nyttig å ha med. Eksempelvis hvis man refererer til noe.

# Hva skal leveres:

* Kildekoden til prototype (Front-End og Back-End)
* Dokumentene i innleveringen skal leveres som PDF-filer, og hele prosjektet  
  (prosjektdokumentasjon, prototype og vedlegg) skal leveres samlet for gruppa som en .zip-fil i Inspera. Det er kun det som er inkludert i gruppeinnleveringen i Inspera som blir tatt med i vurderingen (så ingen eksterne git-repositories, etc.).

Lenke til timeliste:

* <https://1drv.ms/x/c/3b274f0dbc59c8fd/ETfAYYa9NmVOvCu7i6Q7tkkBA8zO2acE7bhsSFaZRmYxpw>

Lenke til Canas dokument:

* <https://hiof.instructure.com/courses/8917/files/1622655?wrap=1>

Lenke til Trello:

* <https://trello.com/b/g7QFBDkP/itf20319-1-24h-software-engineer>
* <https://trello.com/invite/b/66d85ee6de102e85bc54c773/ATTIcdf6677519c556865d398f381dfd7bd800DEA557/itf20319-1-24h-software-engineer>

Lenke til GitHub:

* <https://github.com/EmilB04/S.E.O.T_Gruppe-9>

Lenke til Front-End: