**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

Innholdsfortegnelse

[Introduksjon: Emil 4](#_Toc182679585)

[Problemstilling: Emil 5](#_Toc182679586)

[Løsningsbeskrivelse: Emil 6](#_Toc182679587)

[Kjernefunksjonalitet: 6](#_Toc182679588)

[Åpen og fleksibel flatform 6](#_Toc182679589)

[Data og tilgang 6](#_Toc182679590)

[Ekspanderbarhet og antakelser 7](#_Toc182679591)

[Begrensninger 7](#_Toc182679592)

[Funksjonelle krav: 9](#_Toc182679593)

[Ikke-funksjonelle krav: 10](#_Toc182679594)

[Brukerhistorier og personas (brukssituasjon): 11](#_Toc182679595)

[Systemarkitektur 12](#_Toc182679596)

[Prototypedokumentasjon: 13](#_Toc182679597)

[Testing: 14](#_Toc182679598)

[Utviklingsprosess (må se an denne) 15](#_Toc182679599)

[Instruksjoner for bruk: 16](#_Toc182679600)

[Konklusjon: 17](#_Toc182679601)

[- Modeller og vedlegg: 18](#_Toc182679602)

[Hva skal leveres: 19](#_Toc182679603)

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 smart 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 elbile. 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 eventuelt 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.

**Skriv videre**

# 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

* Forklar løsningen nøye
* Overordnet beskrivelse av systemet som skal utvikles.
* Hvordan løsningen oppfyller kundens behov.
* Begrensninger og antakelser.

For å møte utfordringen i dagens marked med lukkede løsninger for elbillading, har vi 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:

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

## Åpen og fleksibel flatform

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, og ikke bli dratt med 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.
* Integrasjon andre aktører: Muligheten for å utvide med andre aktører, for å gi brukeren flere smarte løsninger når det kommer til lading.

## 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 (med teksttagger eller løpenummer).
* Funksjoner relatert til brukeren: innlogging, varsler, tilgang til strømpriser, etc.
* Gruppekrav basert på funksjonelle områder.
* Forklaring av krav som dekkes i prototypen.

# Ikke-funksjonelle krav:

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

# Brukerhistorier og personas (brukssituasjon):

* Forklar hvordan produktet/løsningen/ideen vår kan bli brukt, og for hvem.

# 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.

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:

* 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>