**Prosjektinnlevering – Gruppe 9**

Emil Berglund

Andreas B. Olaussen

Khalid H. Osman

Sebastian W. Thomsen

Ida K. Tollaksen

Høgskolen i Østfold

Ord: TBD

Emnekode: ITF20319

Innholdsfortegnelse

[Introduksjon: Emil 4](#_Toc182563979)

[Problemstilling: Emil 5](#_Toc182563980)

[Løsningsbeskrivelse: 6](#_Toc182563981)

[Funksjonelle krav: 7](#_Toc182563982)

[Ikke-funksjonelle krav: 8](#_Toc182563983)

[Brukerhistorier og personas (brukssituasjon): 9](#_Toc182563984)

[Systemarkitektur 10](#_Toc182563985)

[Prototypedokumentasjon: 11](#_Toc182563986)

[Testing: 12](#_Toc182563987)

[Utviklingsprosess (må se an denne) 13](#_Toc182563988)

[Instruksjoner for bruk: 14](#_Toc182563989)

[Konklusjon: 15](#_Toc182563990)

[- Modeller og vedlegg: 16](#_Toc182563991)

[Hva skal leveres: 17](#_Toc182563992)

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 oppgaven tar utgangspunkt i en oppstartsbedrift som ønsket å utvikle et smart system for strømvarsling og optimal lading av elbiler hjemme. 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. 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.

# Problemstilling: Emil

* Forklar hvorfor vi valgte å gå for akkurat «denne» ideen.

# Løsningsbeskrivelse:

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

# 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>