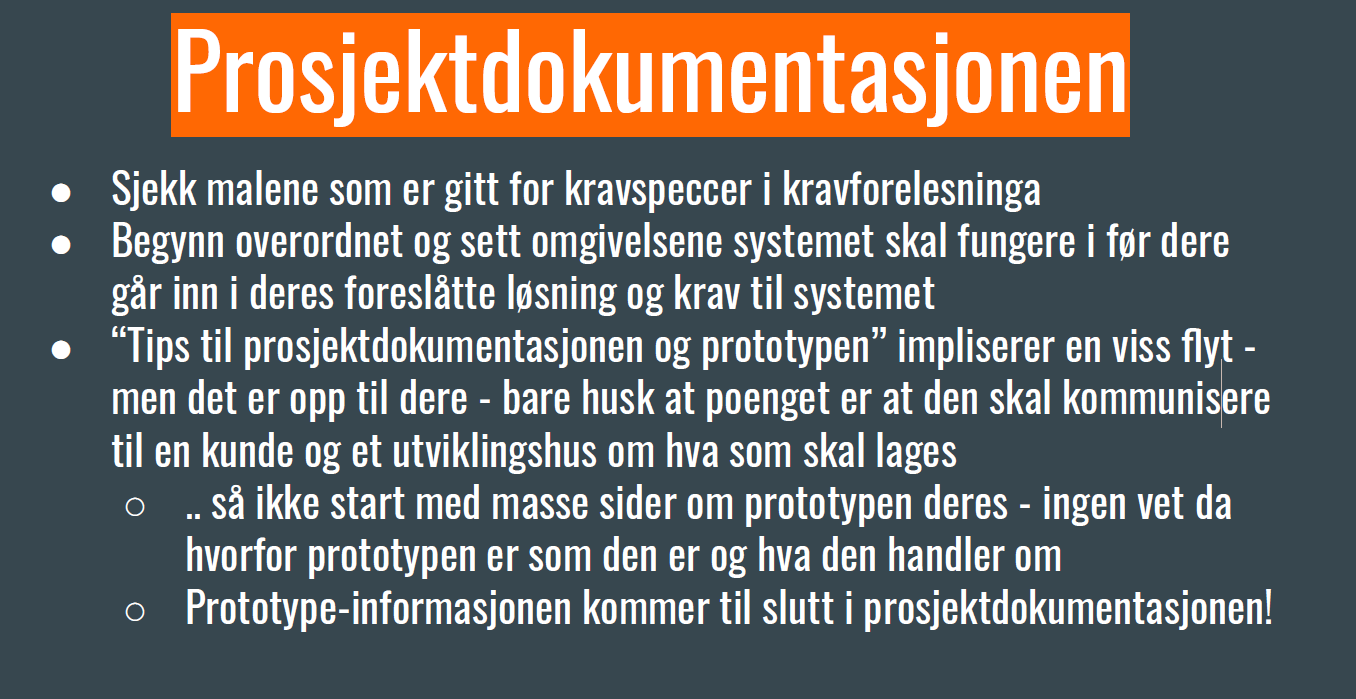
Dokumentasjon plan

* Beskrivelse av problemstillingen og domenet for prosjektet som skal lages.
* Beskriv overordnet hva løsningen til prosjektgruppen går ut på og hvordan løsningen håndterer de forskjellige elementene i problemstillingen.
* Beskriv begrensninger og antagelser dere har gjort i prosjektperioden
* Introduser brukerne av systemet (gjerne beskrevet som såkalte “Personas”) og hvilke oppgaver de ønsker å løse i systemet (f.eks. “Use cases”). Utbrodér de forskjellige brukssituasjonene i mer detaljerte beskrivelser, slik at det er mulig for utenforstående å forstå hva oppgavene som brukerne skal løse i systemet går ut på.
* Introduser brukerne av systemet (gjerne beskrevet som såkalte “Personas”) og hvilke oppgaver de ønsker å løse i systemet (f.eks. “Use cases”).
* .En overordnet modell av prototypen og en beskrivelse av hvordan delene i den hengersammen vil være nyttig for å illustrere hvordan prototypen fungerer. Beskrivelsen skal også vise hvordan prototypen henger sammen med det større systemet dere beskriver.
* Forklar hva som må til for at prototypen deres kan kjøres, og hva en bruker ev. må konfigurere før den kjører.
* Forklar ev. kjente svakheter og problemer med prototypen. Det er bedre at Kunden vet om problemer på forhånd før de ev. merker dem selv.
* gjennom dokumentasjonen og forstå problemet, hva som skal utvikles og hva som er viktig med systemet som utvikles
* Hver person på gruppa skal lage to diagrammer – Disse diagrammene kan være enten dataflytdiagram, sekvensdiagram, tilstandsdiagram eller aktivitetsdiagram. Diagrammene skal vise ulike funksjoner i systemet. Legg inn diagrammene der det er relevant i teksten, ikke for seg selv ved siden av eller til slutt i prosjektdokumentasjonen
* Beskriv gjerne personas og utvidede brukerhistorier i prosjektdokumentasjonen!
* Notér hvilke krav som er implementert i prototypen i prosjektdokumentasjonen - f.eks. ved å bare notere [implementert] på kravene.
  + Lenk kravene til hvilken test som viser at kravet er implementert
  + Denne krav <-> test-relasjonen viser at dere tester viktig funksjonalitet og sørger for at dere passer på testene underveis!
* Lenk gjerne testene til kravene også - // Implements PARKING.PAYMENT.01
* Gruppér kravene etter funksjonsområde - ikke etter bruker
* Antall tester? 15 - 150
* Det første sensor gjerne gjør er å kjøre testene - pass på at INGEN feiler i det det dere leverer (tester skal IKKE feile - om dere forventer feil resultat tilbake, så skal dere sammenligne med det gale resultatet) Testdekning - og husk at ingen tester skal feile
* “Run all tests” i IDE-et hvis ferdig oppsatt - hvis ikke, beskriv i prosjektdokumentasjonen og README.md hvordan testene kjøres!
* Tenk på feilsituasjoner (og test) - slik som at betaling kan feile – kan implementeres ved at dere har egne kortnumre som gir feil dersom dere skal fake betaling i applikasjonen
* 
* Preface
* Introduction
* Glossary
* User reqirements defenition
* System architecture
* System models
* System evolution
* Appendincies
* index