Dokumentasjon oppsett/plan

* Introduksjon
  + Beskrivelse av problemstillingen og domenet for prosjektet.
  + Beskriv overordnet hva løsningen til prosjektgruppen går ut på og hvordan løsningen håndterer de forskjellige elementene i problemstillingen og hva som skal utvikles og hva som er viktig med systemet som utvikles, slik at Kunden og eksterne partnere kan forstå problemet, hva som skal utvikles og hva som er viktig med systemet som utvikles.
  + Beskriv begrensninger og antagelser gjort, slik at det er eksplisitt hvilke rammer prosjektet jobber innenfor
* Glossary(hvis nødvendig)?
* Brukere av systemet
  + Personas/use cases
  + Utbrodere brukersituasjonene slik at det er mulig for utenforstående å forstå hva oppgavene som brukerne skal løse i systemet går ut på.
  + utvidede brukerhistorie
* Avhengigheter brukt i systemet
  + Pass på at dere dokumenterer eksterne avhengigheter, dvs. der systemet må prate med systemer fra andre tjenesteleverandører (f.eks. betalingsløsning), og avgrenser hva systemet skal ha ansvar for (og hva som håndteres av eksterne partnere).
  + Forklare hvilke Avhengigheter som brukes i applikasjonen, og at disse installeres automatisk ved hjelp av pakkesystem
* Kravspecc
  + Kravliste med nummer eller navn
    - Noter hvilke krav som er implementert i prototypen - 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
    - Grupper kravene etter funksjonsområde - ikke etter bruker
    - Antall tester? 15 – 150
  + Kravene til systemet skal være dokumentert på et slikt nivå at dokumentasjonen kan sendes til eksterne selskap som kan utvikle, teste og evaluere systemet i samarbeid med produkteier og oppdragsgiver.
  + Ha med estimat-tabell for utvikling størrelse og nytte (allerede laget)
  + Kravene bør spore hvor i prototypen de er testet (f.eks. med ved å referere til testnavnet og filnavnet som testen ligger i), slik at det er enkelt å ha oversikt over hva prototypen inneholder og som er implementert.
  + Henvis til kravene som er nevnt i prosjektdokumentasjonen fra testene deres (“/\* Tester krav …. \*/”) og motsatt (i prosjektdokumentasjonen: “Testes i xyz/foo.java”)
* Kom i gang med systemet
  + Forklare hva som må til for at prototypen deres kan kjøres, og hva en bruker ev. må konfigurere før den kjører.
  + Forklare hvordan kjøre tester - “Run all tests” i IDE-et hvis ferdig oppsatt - hvis ikke, beskriv hvordan testene kjøres!
  + hvordan en person uten dyptgående IT-kunnskap kan bygge, kjøre og teste prototypen deres. Bør være “åpne prosjektet i IDE-et, trykk kjør”.
  + –
* Beskrivelse av systemet (dette skal sent i dokumentasjonen – ikke først)
  + En overordnet modell av prototypen og en beskrivelse av hvordan delene i den henger sammen for å illustrere hvordan prototypen fungerer.
  + Beskrivelsen skal også vise hvordan prototypen henger sammen med det større systemet dere beskriver
  + Forklar ev. kjente svakheter og problemer med prototypen. Det er bedre at Kunden vet om problemer på forhånd før de ev. merker dem selv.
  + Systemet som helhet skal være dokumentert på et slikt nivå at dokumentasjonen kan sendes til eksterne selskap som kan utvikle, teste og evaluere systemet i samarbeid med produkteier og oppdragsgiver
  + Det er ikke nødvendig med en fullstendig oversikt over systemet på detaljnivået, men viktige og kompliserte deler av systemet bør beskrives overordnet slik at det er tydelig hvordan systemet er tenkt til å fungere
  + Vis frem ting fra prototypen slik at Kunden får en forståelse for hvordan problemstillingen er løst, uten at de må kjøre prototypen på egen maskin (f.eks. med skjermbilder og beskrivelser). Dette er ikke en sluttbrukermanual, men en forklaring til hvordan dere har prototypet de viktigste kravene.
* DELE OPP I FLERE DELER OM SYSTEMET - boka
* Systemets utvikling
  + Foretelle om hvordan vi har bygget opp systemet, hva vi har

Ting å huske på/sjekke:

* 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
* 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
* 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
* Det skal også lages nødvendige tester som viser at prototypen tilfredsstiller de viktigste kravene som dere har avdekket og som blir beskrevet i dokumentasjonen dere skal produsere for systemet.
* Dere skal ikke integrere mot noen eksterne leverandører i prototypen, men bør skrive små “stubs” som gir dere muligheten til å teste funksjoner som avhenger av dem.
* Husk git mappa i innlevering
* Avhengigheter i applikasjonen deres skal kunne installeres automatisk ved hjelp av et pakkesystem for språket eller rammeverket dere.
* IKKE BRUK TID PÅ:
  + Innlogging systemer – lag ferdig knapp eller dropp det
  + Fancy brukergrensesnitt
* MYE viktigere å ha mange gode tester som tester alt som kan testes
* Til slutt: Prøv hele systemet på en ny maskin og se at alt vil installeres og fungere godt hos sensor
* Ikke fast sidetall: 20-60 sider, og 15-100 tester
* Husk at prosjektdokumentasjonen skal kommunisere med folk som IKKE er IT-folk eller utviklere også! Den som leser dokumentasjonen har ikke samme domenekunnskap som prosjektgruppa besitter etter å ha jobbet med problemstillingen. Sørg for å forklare nødvendige elementer for å unngå tvetydighet og misforståelser
* Prosjektdokumentasjonen skal være skrevet slik at utenforstående kan lese gjennom den og forstå problemet, hva som skal utvikles og hva som er viktig med systemet som utvikles. Husk at de som skal lese dokumentasjonen ikke nødvendigvis kjenner problemstillingen eller prosjektgruppens forslag til løsning fra tidligere.
* Sett figurer, tabeller og diagrammer i kontekst - pass på at dere sier tydelig hva de viser og hva de ev. ikke viser, og hva nytten og konklusjonen av materiellet er.
* Husk forklaringer til tabeller og diagrammer
* Test instruksjonene på en maskin som ikke er brukt til å utvikle prosjektet. Husk at dette også gjelder hvordan testene til prototypen kan kjøres.
* Pass på at testene deres faktisk tester funksjonene! Dersom dere endrer ting inne i funksjonene som testes på en slik måte at resultatet blir galt, så skal testen brekke. Det hjelper ikke om testen alltid passerer (“blir grønn”) - da er det ikke en test.