**Definér kodestandarder**

**Solution**Der bliver oprettet en solution, kaldet "LevelUp". Denne solution indeholder de følgende projekter:  
DataAccess,   
Controller  
RESTService  
Client  
UnitTest

De fire første sikrer en fornuftig lagdeling af arkitekturen, med DataAccess som modellag, Controller, som controllerlag, RESTService som servicelag og Client som klientlag. Da der er tale om en ASP.Net klient, og da denne har indbygget et specifik arkitektur, benytter vi MS standard. Det sidste projekt benytter vi i forbindelse med Test First-udviklingsprocessen, og alle metoder til test opbevares heri.

**Klasser**Alle klasser skrives som UpperCamelCase. I modellaget er der blot et sigende navn, uden specifik endelse. Kontrollaget slutter alle klasser med "Controller", mens alle interfaces i servicelaget starter med stort "I" efterfulgt af modelnavnet. [OI! Ikke log!! fix it]  
I klienten har hver side navn efter dens primære funktion, og derfor må den godt adskille sig fra modelnavnene. Så længe det er klart hvilken primær model den benytter.  
f.eks. CreateUser for siden til at oprette bruger.  
I App\_Code benyttes endelsen "Calls" på de klasser der har til ansvar at hidkalde servicen. Mens hjælpeklasser har endelsen "Helper"

**Metoder**Alle metoder skrives som UpperCamelCase. Vi benytter .Net inline properties til get/set på instansvariabler.   
På metoder åbnes og lukkes tuborgklammerne under metodeheaderen. På kontrolblokke åbnes klammerne under statement, og afsluttes under.

Metoderne skal have sigende navne. Hvis man søger et objekt, men ikke laver ændringer til det, skal metoden starte med "Get". Hvis man laver en ændring skal metoden starte med "Update". Ved oprettelse af et objekt skal den starte med "Create" og ved sletning skal det være "Delete".  
I tilfælde der falder uden for CRUD skal der anvendes Sund Fornuft ved oprettelsen.

**Variabler**  
Alle almindelige variabler skrives lowerCamelCase, mens konstanter skrives STORT\_MED\_UNDERSTREGNING.