

## **FURPS+ Gennemgang**

### **Functionality:**

Inden udarbejdelse af projektet blev de funktionelle krav formuleret til user stories. Disse er fremgået på følgende måde:

1. Som bruger vil jeg gerne kunne danne mig et overblik over artists
2. Som bruger vil jeg gerne kunne se detaljer om en bestemt artist
3. Som bruger vil jeg gerne kunne filtrere og sortere artists på udvalgte properties
4. [...] oprette en ny artist
5. [...] slette en artist
6. [...] opdatere en artists oplysninger
7. [...] tilføje og fjerne en artist som favorit
8. [...] danne mig et overblik over favorit-artists

Applikationen opfylder samtlige funktionelle krav, herunder CRUD-operationer, filtrering og sortering samt præsentation af artist-data. De funktionelle krav er implementeret på en robust og pålidelig måde, som jeg vil uddybe i de kommende afsnit omhandlende non-funktionelle krav.

### **Usability:**

Applikationen overholder de fire ”golden rules” inden for UI-design og gestaltlovene, hvilket gør den brugervenlig og intuitiv. Loven om nærhed opfyldes bl.a. gennem nærheden af ”Update artist” og ”Delete artist” nærhed til hinanden i details-view samt også nærheden af filtrering- og sorteringsfeltet øverst på siden. Loven om lighed overholdes ved ligheden af de flere artist-items ensartethed ifm. form, farve, størrelse og mønster. Loven om lukkethed opfyldes bl.a. gennem artist-items og dialog-elementet mørkere farve sammenlignet med baggrunden. Loven om forbundethed opfyldes ved samtlige formularer udført under CRUD-operationer, hvor beskrivelsen af input-feltet og selve feltet er designet således, at de er tæt forbundet.

### **Reliability:**

Stabilitet og pålidelighed vægtes højt i applikationen. Typescripts mange sikkerhedsnet under implementeringsfasen har spillet en afgørende rolle for at sikre nævnte.

### **Performance:**

Applikationen er udviklet således, at der ikke hentes data mere end højest nødvendigt. Dette er blevet besluttet mhp. at opnå en hurtig ydeevne og hurtig datahentning.

### **Supportability:**

Koden er struktureret i moduler, hvilket vil gavne fremtidige udviklere i at forstå og videreudvikle applikationen. Derudover er koden dokumenteret med in-line dokumentation mhp. at sikre en gnidningsfri overdragelse af koden til andre udviklere.

+:

Applikationen overholder de generelle principper for kodestruktur. Separation of Concerns er opnået gennem adskillelse af koden i forskellige moduler samt FOP (funktionsorienteret programmering). Funktionerne har højst kun afhængigheder en vej (f.eks. ved søgning, filtrering og derefter sortering), hvorfor loose coupling er opnået. Til sidst har jeg opnået high cohesion ved at sikre, at funktioner, der arbejder om at udføre et bestemt funktionelt krav, er samlet som tæt som muligt bl.a. gennem closures og moduler.