

- **Functionality:** Applikationen skal opfylde kravene for CRUD-operationer, filtrering, sortering og præsentation af kunstnerdata. Funktionaliteten skal være pålidelig og intuitiv.
  - **Crud:** Create, Read, Update og Delete er alle tilgængelig igennem frontend og backend api.
  - **Filtrering:** Ved at klikke på "favorite" knappen vil man filtrere igennem ens favorit kunstnere
  - **Sortering:** Man kan sortere efter alfabetisk rækkefølge og man kan søge.
  - **Kunstner data:** Vist som en liste af navne på kunstnerne som man kan klikke på for at få detaljer om kunstneren.
- **Usability:** Brugergrænsefladen skal være brugervenlig og nem at navigere. Alle CRUD-operationer skal være let tilgængelige, og filtrerings/sorteringsfunktionerne skal være intuitive.

- **Brugergrænsefladen:** Alle knapper er navngivet for at gøre det mere bruger venligt, Kunstner dataen er mindre brugervenlig men meget nem efter man har forstået den.
- **CRUD:** Alle CRUD-operationer er nemme at bruge og simple.
- **Reliability:** Applikationen skal være stabil og pålidelig. Dataintegritet og korrekt håndtering af CRUD-operationer er afgørende.
  - API 'en er stabil ved at den har meget bestemte krav om hvad der skal med i Operationer, derudover har den try/catch med hjælpsomme errors den sender tilbage.
- **Performance:** Applikationen skal have en acceptabel ydeevne, herunder hurtig datahentning og responsivitet i brugergrænsefladen.
  - Applikationen er hurtig til API calls. Hvilket man kan se fx når man sletter en kunstner.
- **Supportability:** Koden skal være velstruktureret og veldokumenteret, så det er nemt for fremtidige udviklere at vedligeholde og udvide applikationen.
  - I koden er der kommentere og sat op i funktioner med navne der forklarer hvad funktionen gør.
- **+**: Der skal også tages hensyn til andre relevante krav og overvejelser for at sikre en vellykket udvikling og implementering af applikationen. Ud over FURPS-kravene skal udviklingen også overholde generelle principper for kodestruktur, modularitet og afhængighedsstyring, herunder Separation of Concerns, Loose Coupling og High Cohesion.
  - Funktionerne er så uafhængige af hinanden som virkede muligt, de fleste funktioner er dog linked med API som betyder at hvis man ændrer krav til hvad der skal være i et api call ville flere funktioner skulle ændres.