

**Use Case:** Præsentation af Musikinformation på MusicBase

**Hovedaktør:** Bruger

**Mål:** At søge og præsentere information om kunstnere, sange og albums ved hjælp af objekt-orienterede programmeringsprincipper.

**Forudsætninger:**

- Brugeren har adgang til MusicBase-webapplikationen.
- Databasen indeholder information om kunstnere (artists), udgivelser (albums) og enkelte numre (tracks).

**Hovedscenarie:**

- Brugeren åbner MusicBase-webapplikationen i en webbrowser.
- Brugeren ser et brugergrænseflade (UI) opdelt i forskellige søgemuligheder og præsentationer af musikinformation.
- Musikopslagssiden indlæses, og UI-komponenterne initieres som JavaScript-objekter.
- Brugeren vælger en søgefunktion, f.eks. "Søg efter kunstnere."
- Musikopslagssiden bruger objekt-orienterede komponenter til at håndtere brugerens anmodning.
- MusicBase-applikationen udfører en søgning og henter information om kunstnere fra databasen.
- De hentede data præsenteres som objekter, som indeholder metoder og egenskaber til at vise informationen.
- Brugeren ser en liste over kunstnere, hvor hvert element repræsenterer en kunstner som et objekt.
- Brugeren kan også foretage søgninger efter sange og albums, og resultatet præsenteres i en lignende objekt-orienteret struktur.

**SSD Diagram:** Præsentation af Musikinformation på MusicBase

**Actor:** Bruger

**System:** MusicBase-webapplikation

**Use Case:** Præsentation af Musikinformation

**Hovedscenarie:**

- Bruger: Åbner MusicBase-webapplikationen.
- System: Initialiserer og viser grænsefladen (UI).
- Bruger: Vælger "Søg efter kunstnere" fra UI'en.
- System: Modtager anmodning om at søge efter kunstnere.

System: Udfører en søgning i databasen efter kunstnere.

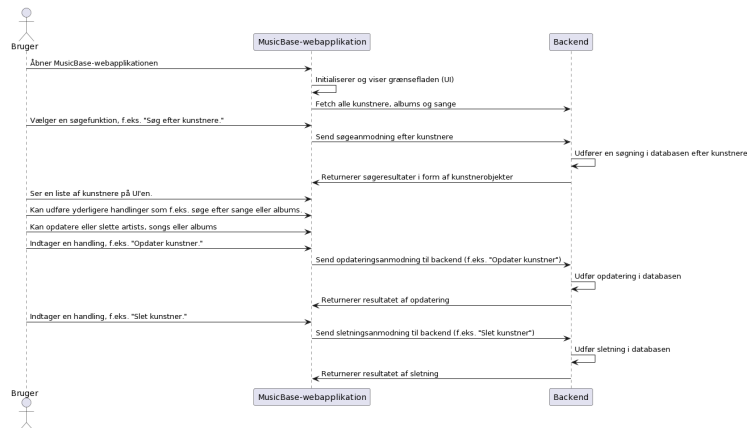
System: Returnerer søgeresultater i form af kunstnerobjekter.

Bruger: Ser en liste af kunstnere på UI'en.

Bruger: Kan udføre yderligere handlinger som f.eks. søge efter sange eller albums.

Bruger: Kan opdatere eller slette artists, songs eller albums

SD:



Evt snak om normalformer

