

Backend Software Development

Technisch Ontwerp

Student: Rick de Jong

Docent: Robin Bakels

Opleider: NOVI

25-05-2022



NOVI
HOGESCHOOL

Inhoudsopgave

| | |
|---------------------------|---|
| Opdrachtbeschrijving..... | 3 |
| Systeemeisen..... | 3 |
| User Stories..... | 3 |
| Klassendiagram..... | 4 |
| Sequentiediagrammen..... | 5 |

Opdrachtbeschrijving

Voor mijn eindopdracht zet ik een backend systeem op voor een muziekwinkel. Er is mij gevraagd een systeem te ontwerpen waarmee klanten een overzicht kunnen krijgen van albums, artiesten en labels welke de winkelmedewerkers kunnen manipuleren.

Een admin-account kan artiesten toevoegen aan een label en albums toevoegen aan een artiest – wanneer een album wordt aangemaakt dan slaat het de huidige label van de artiest op als eigen label, welke blijft staan zelfs als de artiest van label wisselt.

Systeemeisen

Een user-account moet in staat zijn een lijst met opgeslagen albums te zien, evenzo een van artiesten en een van labels, maar moet deze niet kunnen aanpassen.

Tevens moet een user in staat zijn te zoeken op een specifieke artiest of label, en alleen hier de gegevens van te tonen.

Een admin-account moet in staat zijn de bovengenoemde functies te gebruiken, maar ook verschillende bewerkingen uit te voeren zoals:

- Nieuwe albums/artiesten/labels toevoegen;
- Albums/artiesten/labels verwijderen;
- De naam aan te passen van een album/artiest/label;
- Een artiest (ont)koppelen aan een label;
- Een album aan een artiest te koppelen

User Stories

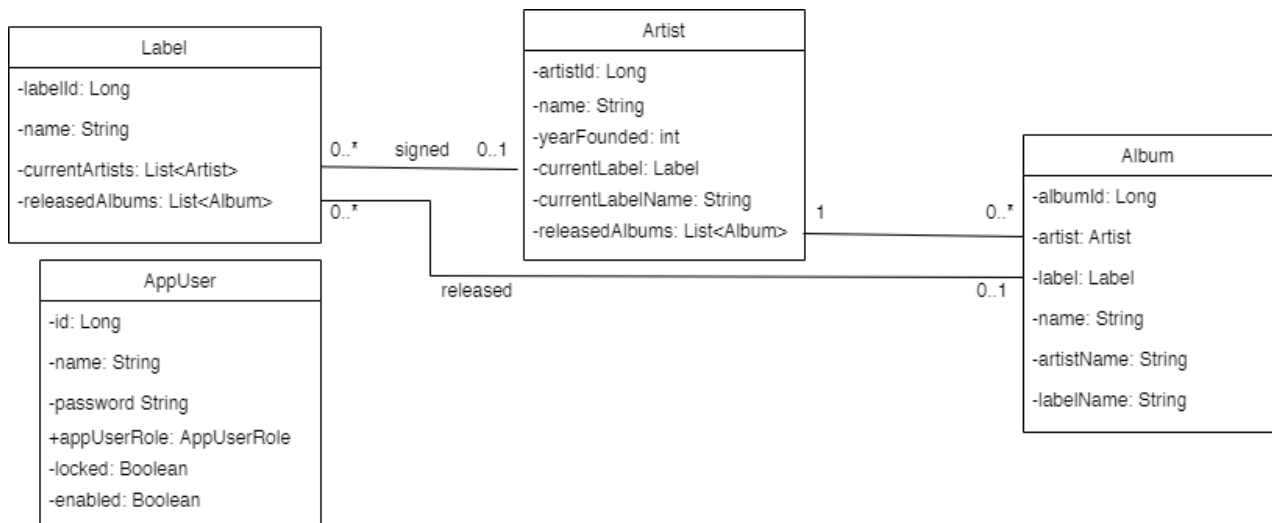
Als klant wil ik graag een overzicht van alle albums van een artiest om te kunnen zien welke ik nog mis in mijn collectie.

Als klant wil ik kunnen zien welke artiesten bij dezelfde label getekend hebben om soortgelijke acts te ontdekken.

Als administrator wil ik nieuwe artiesten/albums/labels kunnen toevoegen aan de database, met de daarbijhorende relevante informatie, zodat klanten een vollediger overzicht kunnen krijgen.

Als administrator wil ik de namen van artiesten/albums/labels kunnen aanpassen om mogelijke foutjes bij het aanmaken te herstellen.

Klassendiagram



Toelichting:

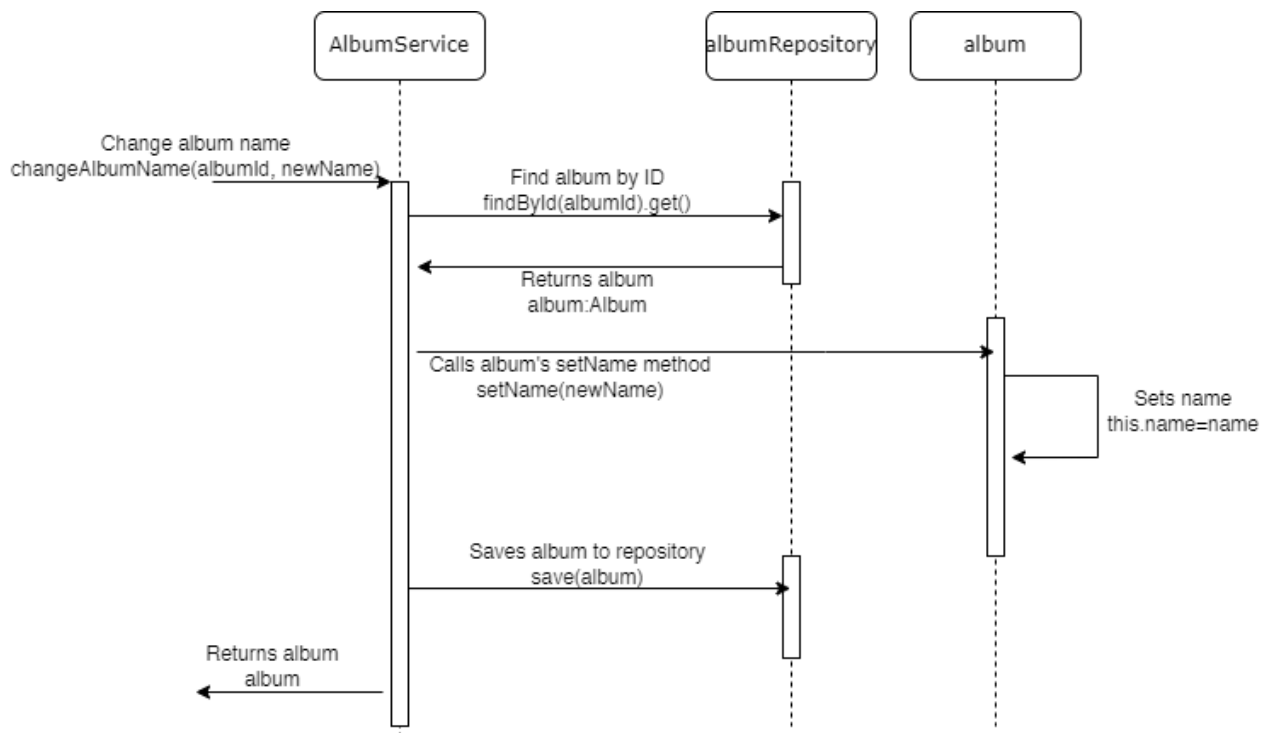
In dit diagram worden de relaties tussen de vier hoofdklassen weergegeven. Een artiest kan getekend hebben bij een label, maar altijd maar bij één tegelijk, of onafhankelijk zijn. Tevens kan een artiest nul tot meerdere albums uitgegeven hebben, maar een album moet altijd één artiest hebben. Een album neemt op het moment van aanmaken de label over van de artiest, en behoudt deze zelfs wanneer de artiest van label wisselt. Als de artiest geen label had op het moment van aanmaken zal deze worden opgeslagen als 'independent'.

De AppUser heeft geen directe relatie met de andere drie klassen maar houdt zich bezig met de manipulatie van de database. Hierin worden gegevens van de gebruiker opgeslagen als diens naam, het wachtwoord en de rol, die de rechten van de gebruiker bepaalt.

Voor een volledig overzicht van *alle* klassen is er een diagram beschikbaar op mijn github: <https://github.com/SilvenV/Muziekwinkel/blob/master/Muziekwinkel%20UML.png>

Sequentiediagrammen

In het volgende sequentiediagram wordt weergegeven welke stappen het programma onderneemt wanneer een gebruiker met de ADMIN rol de naam van een album wil aanpassen.



In het volgende sequentiediagram wordt weergegeven welke stappen het programma doorloopt wanneer er een artiest toegevoegd (signed) wordt aan een label. Gezien de grootte van het diagram wordt deze zijwaards weergegeven.

