

INHOUD

Opdrachtsomschrijving	2
Productvisie en -goal	2
Vastgesteld Klassendiagram	4
Klassenomschrijving	5

OPDRACHTSOMSCHRIJVING

Maak een C# CLI (Command Line Interface) applicatie en realiseer de “Spotify” applicatie.

Het project dient gedocumenteerd te worden via GitHub. Maak een SCRUM-board aan op de GitHub (zie tab “Projects” binnen GitHub). Vul een sprintbacklog met de eisen en wensen. Alle user stories in de sprintbacklog dienen te voldoen aan de criteria: **Wie, Wat, Waarom** en **Realistisch**. Iedere user story dient een duidelijke **Definition of Done** te bevatten.

Vanuit de eisen en wensen dient een ontwerp gemaakt te worden. Verifieer de functionaliteit van het **vastgesteld klassendiagram** doormiddel van activiteitendiagrammen of use-case diagrammen.

Na het realiseren dient de applicatie **doorgroondig getest** te worden. Alle testcases dienen aan te sluiten op de user stories en bevatten **alle** scenarios. Alle stappen, het gewenste resultaat en testdata zijn benoemd. Het is belangrijk dat niet alleen het hoofdscenario, maar ook alternatieve scenarios worden getest.

Alle resultaten en conclusies van het testen dienen omschreven te worden in een **testrapport**.

Tot slot wordt er een **verbetervoorstel** gedaan richting de **product-owner**. De verbetervoorstellen dienen gedaan te worden vanuit het testen, de oplevering en een reflectie. Het is belangrijk dat je alle opvallende details documenteert en advies uitbrengt hoe het project succesvol kan worden overgedragen.

Tot slot moet de **product-owner** worden overtuigd dat het werk **authentiek** is. Toon op zoveel mogelijke manieren aan dat je begrijpt wat je hebt gemaakt.

PRODUCTVISIE EN -GOAL

Hendrik Kavel, een rijke entrepreneur, wil graag een concurrent maken voor Spotify. Hij heeft opgemerkt dat Spotify, ondanks veel goede kwaliteiten, toch wat specifieke functionaliteit mist. Hij heeft ervoor gekozen om jullie in te huren om zijn eigen muziekclient na te maken. Hendrik wil uiteindelijk dat zijn muziekclient echt muziek kan afspelen, maar voor nu vind hij het prima als zijn muziekclient alleen via tekst de functionaliteit laat zien.

De applicatie hoeft nog geen verbinding te maken met een database, het dient voor nu alleen als een platform om nieuwe functionaliteit te testen. Alle nummers, albums en gebruikers mogen dus hard-coded aangemaakt te worden. Binnen de applicatie dient een hoofdgebruiker gekozen te worden. Alle acties die worden ondernomen, worden dan ook vanuit deze hoofdgebruiker uitgevoerd. Dus bijvoorbeeld: als een nieuwe speellijst wordt aangemaakt, dan is deze automatisch van de hoofdgebruiker.

Speellijsten:

Een gebruiker kan meerdere speellijsten hebben en moet nummers, albums en andere speellijsten kunnen toevoegen aan een speellijst. De gebruiker moet alle nummers van een specifieke speellijst kunnen inzien. Ook moet de gebruiker een nummer uit de speellijst kunnen verwijderen. Tot slot moet de gebruiker een speellijst moeten kunnen afspelen. Bij het afspelen van een speellijst, begint het eerste nummer in de lijst te spelen. Wanneer het laatste nummer van een speellijst is gespeeld, dan stopt het spelen automatisch. Wanneer een nummer uit een speellijst wordt overgeslagen, dan moet het volgende nummer beginnen met spelen.

Gebruikers:

Een hoofdgebruiker moet kunnen inzien welke andere gebruikers deel zijn van de nieuwe spotify client. -Van deze andere gebruikers moet de hoofdgebruiker iemand kunnen toevoegen als vriend. Ook moet de hoofdgebruiker een vriend kunnen verwijderen. De hoofdgebruiker moet alle vrienden kunnen inzien. De hoofdgebruiker moet speellijsten van vrienden kunnen inzien. Dit betekent dat er een lijst van speellijsten naar voren moet komen. De hoofdgebruiker moet vervolgens alle nummers van een specifieke speellijst kunnen inzien. Tot slot, de hoofdgebruiker moet zo'n geselecteerde speellijst kunnen overnemen of toevoegen aan een eigen speellijst.

Albums:

Albums hebben een titel, een verzameling van artiesten en een verzameling van nummers. Aan albums kunnen geen nieuwe nummers worden toegevoegd. Een hoofdgebruiker moet alle nummers of albums kunnen inzien. Dit betekent dat er een lijst van alle nummers of albums naar voren moet komen. Uit deze nummers of albums moet de gebruiker nummers kunnen selecteren. De gebruiker kan op deze wijze een nummer of album toevoegen aan een eigen speellijst, of kunnen afspelen. Het afspelen van een album gebeurt op dezelfde manier als een speellijst.

Nummers:

Nummers hebben een titel, een verzameling aan deelnemende artiesten en een genre. Wanneer een nummer wordt afgespeeld dienen deze geschreven te worden naar de console. Er hoeft niet een daadwerkelijk nummer te afgespeeld te worden. Wanneer een nummer wordt overgeslagen, dan begint hetzelfde nummer te spelen. Wanneer een nummer wordt gestopt, dan zal de console tonen dat het huidige nummer is gestopt met spelen. Het pauzeren van een nummer hoeft voor het prototype nog niets te doen.

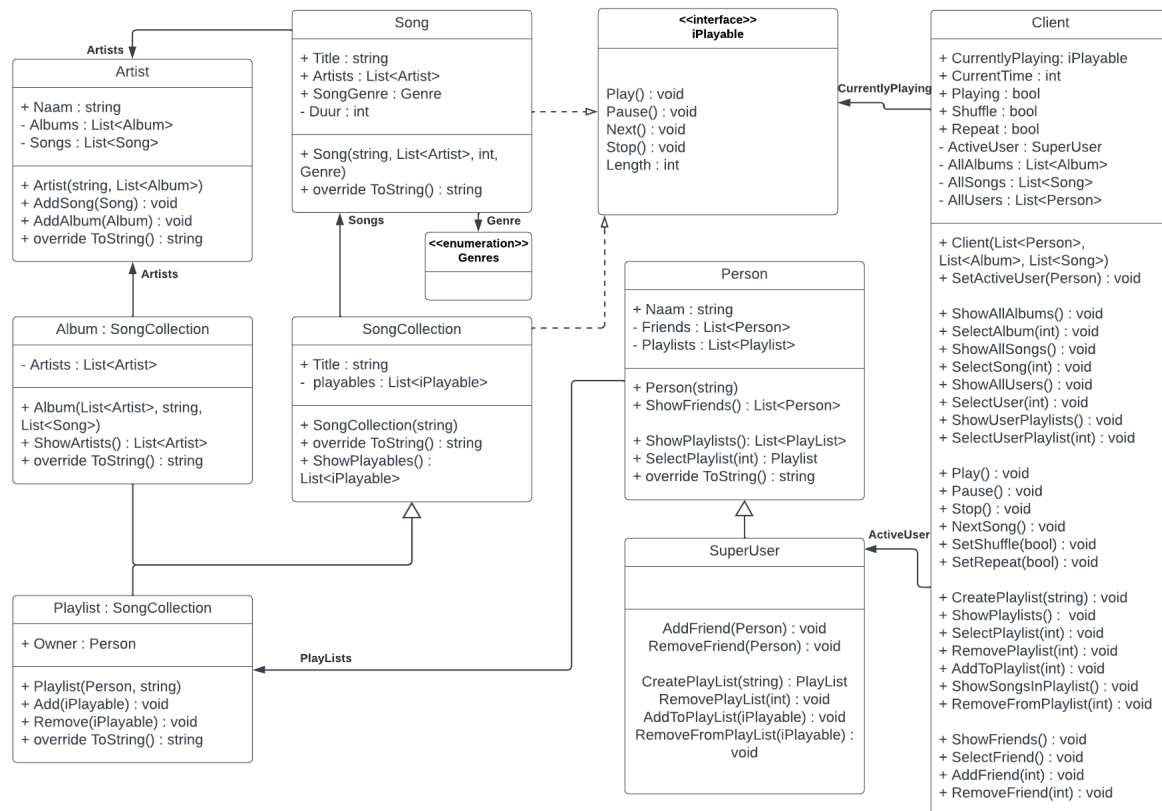
Artiesten:

Artiesten hebben een naam en verzamelingen van albums en nummers waar zij aan hebben gewerkt. De hoofdgebruiker moet een alle artiesten kunnen inzien. Dit betekent dat alle albums en nummers (singles) waar een artiest aan heeft gewerkt naar voren moet komen.

Client:

Alle CLI interactie dient via de Client class afgehandeld te worden. Dus bijvoorbeeld: Wanneer de gebruiker een nummer wil selecteren, dan dient de daarbij horende methode van Client aangeroepen te worden.

VASTGESTELD KLASSENDIAGRAM



Figuur 1: Klassendiagram Spotify (UMLs)

KLASSENOMSCHRIJVING

CLASS	METHOD	OMSCHRIJVING
CLIENT	ShowAllAlbums()	Toont alle bestaand albums binnen de client
CLIENT	SelectAlbum(int)	Selecteer een bestaand album binnen de client
CLIENT	ShowAllSongs()	Toont alle bestaand nummers binnen de client
CLIENT	SelectSong(int)	Selecteer een bestaand nummer binnen de client
CLIENT	ShowAllUsers()	Toont alle bestaande gebruikers binnen de client
CLIENT	SelectUser(int)	Selecteer een bestaande gebruiker binnen de client
CLIENT	ShowUserPlaylist()	Toon alle bestaande afspeellijsten van een gebruiker binnen de client
CLIENT	SelectUserPlaylist(int)	Selecteer een bestaande afspeellijst van een beschikbare gebruiker binnen de client
CLIENT	Play()	Speel de huidig geselecteerde iPlayable af
CLIENT	Pause()	Pauzeer de huidig geselecteerde iPlayable
CLIENT	Stop()	Stop de huidige geselecteerde iPlayable
CLIENT	Next()	Doorgaan naar het volgende beschikbare nummer. Eén enkele nummer wordt opnieuw afgespeeld
CLIENT	SetShuffle()	Zet shuffle aan/uit
CLIENT	SetRepeat()	Zet repeat aan/uit
CLIENT	CreatePlaylist(string)	Actieve gebruiker maakt een nieuwe afspeellijst aan
CLIENT	ShowPlaylists()	Toon alle bestaande afspeellijsten van de actieve gebruiker
CLIENT	SelectPlaylist(int)	Selecteer een bestaande afspeellijst van de actieve gebruiker
CLIENT	RemovePlaylist(int)	Verwijder een bestaande afspeellijst van de actieve gebruiker
CLIENT	ShowSongsInPlaylist()	Toon alle bestaande nummers binnen de huidig geselecteerde afspeellijst
CLIENT	AddToPlaylist(Playable)	Voeg een nieuwe iPlayable toe aan de huidig geselecteerde afspeellijst
CLIENT	RemoveFromPlaylist(int)	Verwijder een nummer van de huidig geselecteerde afspeellijst
CLIENT	ShowFriends()	Toon alle vrienden van de actieve gebruiker
CLIENT	SelectFriend()	Selecteer een gebruiker uit de vriendenlijst van de actieve gebruiker
CLIENT	AddFriends(int)	Voeg een bestaande gebruiker toe aan de vriendenlijst van de actieve gebruiker
CLIENT	RemoveFriend(int)	Verwijder een gebruiker van de vriendenlijst van de actieve gebruiker

CLASS	METHOD	OMSCHRIJVING
SONG	Play	Laat via de console zien dat het nummer wordt afgespeeld
SONG	Pause	Laat via de console zien dat het nummer is gepauzeerd
SONG	Stop	Laat via de console zien dat het nummer is gestopt
SONG	Next	Roept Play aan van zichzelf
SONG	ToString	Schrijft de titel + naam van de artiest + lengte naar de console
SONGCOLLECTION	Play	Doet een aanroep naar Play van de actieve Song
SONGCOLLECTION	Pause	Doet een aanroep naar Pause van de actieve Song
SONGCOLLECTION	Stop	Doet een aanroep naar Stop van de actieve Song
SONGCOLLECTION	Next	Maakt de volgende Song de actieve Song
PERSON	ToString	Geeft de naam + vrienden + speellijsten terug als String
PERSON	ShowFriends	Schrijft één voor één de vrienden van de persoon naar de console
PERSON	ShowPlaylists	Schrijft één voor één de naam van de afspeellijsten van de persoon naar de console