|  |  |
| --- | --- |
| **Project title** | **Music Player** |
| **Author(s)** | **Zară Antonia-Maria** |
| **Group** | **30423** |

Music Player

1. Description

I built a Java Application for a Music Player. The application is a wannabe Spotify player, where the user can buy songs or albums, play them, follow artists and befriend people, therefore being both a music player and a social media app.

1. Classes
   1. User

* Represents the super class for the three types of users the app will include
* It contains universal personal details about the users, such as username, password, email, first name, last name, email, userId, country, birthday, status
  1. StandardUser
* Represents the first type of user, which has the most methods implemented due to the limiting nature of the app on this type of user
* It contains a playlist array, an array of owned songs, an array of owned albums, a user type, an array of friends which are standard users, an array of friends which are premium users and an array of following artists
  1. PremiumUser
* Represents the new tier of user type
* It contains the same arrays of friend as the standard user, an array of following artists, an user type and a playlist array
  1. Artist
* Represents the last type of user
* It contains a playlist array, an album array, an user type and an array of users who follow this artist
  1. Song
* Points to an artist and an album along other details
  1. Album
* Extends the Playlist class with fields for an artist and a genre
  1. Playlist
* Contains an user creator and an array of songs among other details
  1. Genre
* Enum for defining music genres for the albums
  1. Countries
* Enums for defining all possible countries a user can be from
  1. UserType
* Enum for defining all three types a user can be

A picture containing text, sky, map

Description automatically generated

1. Features

* There are three types of users
* Standard and premium users can befriend and unfriend any other standard or premium users, and before doing so they can see their details next to their names in a separate panel
* Artists are not allowed to befriend any user
* Standard and premium users can follow artists as a sign of appreciation
* Artists are not allowed to follow another artist
* Standard users can buy songs and albums and add them to their library
* Premium users and artists cannot buy songs or albums because they will all be available for them in their library
* All users can add either an unlimited number of songs or albums in their song queue for later play
* Music can be played in two modes: ordered, which stops the play after the last song has been played, or shuffle, where the program picks a random song from the queue and plays them without limit
* Any user can see their status in the personal panel
* Any user can change their status
* A standard user can upgrade to a premium user
* A premium user can upgrade to an artist
* An artist can see all his albums, songs
* An artist can see his followers

Technical Features:

* Each button from the play, buy or people panels checks if anything is selected, if not, displays an informative message
* If an user type presses a button that represents limited access for that user type, the program displays an informative message
* Music can be played from a selection of .wav files stored in the project folder, using a Swing feature implementing the Clip file using the MusicStuff Class
* Display information is changed real-time after any internal change
* All the data concerning the main classes is taken from a MySQL database implemented in IntelliJ, alongside the project, and if necessary, the program inserts, deletes or updates entries as well using the Database Class

GUI:

* I used Swing without Scene Builder in order to build the User Interface of my application
* I have two main frames, one for the login window which selects the user that the whole app behaviour depends on, and the second for the main app activities
* The graphic user interface includes many panels, each panel being sketched in a separate class, described below:

Graphical user interface, text, application, email

Description automatically generated

* ArtistPanel: consists of DeleteSongSplitPane, DeleteAlbumSplitPane, FollowerSplitPane

(updated)