Final Report

Project team name: 06 MusicGeneration
Team member: Nick Montoya
Yifan Li
Vision: Allow users to create, find, and auto-generated playlists based on factors of their choice

Description: Django based web application that searches Spotify's API for songs similar to a pre-existing playlist in the users library. Users are able to select and add their favorites from the generated playlists.

List the features that were implemented (table with ID and title)

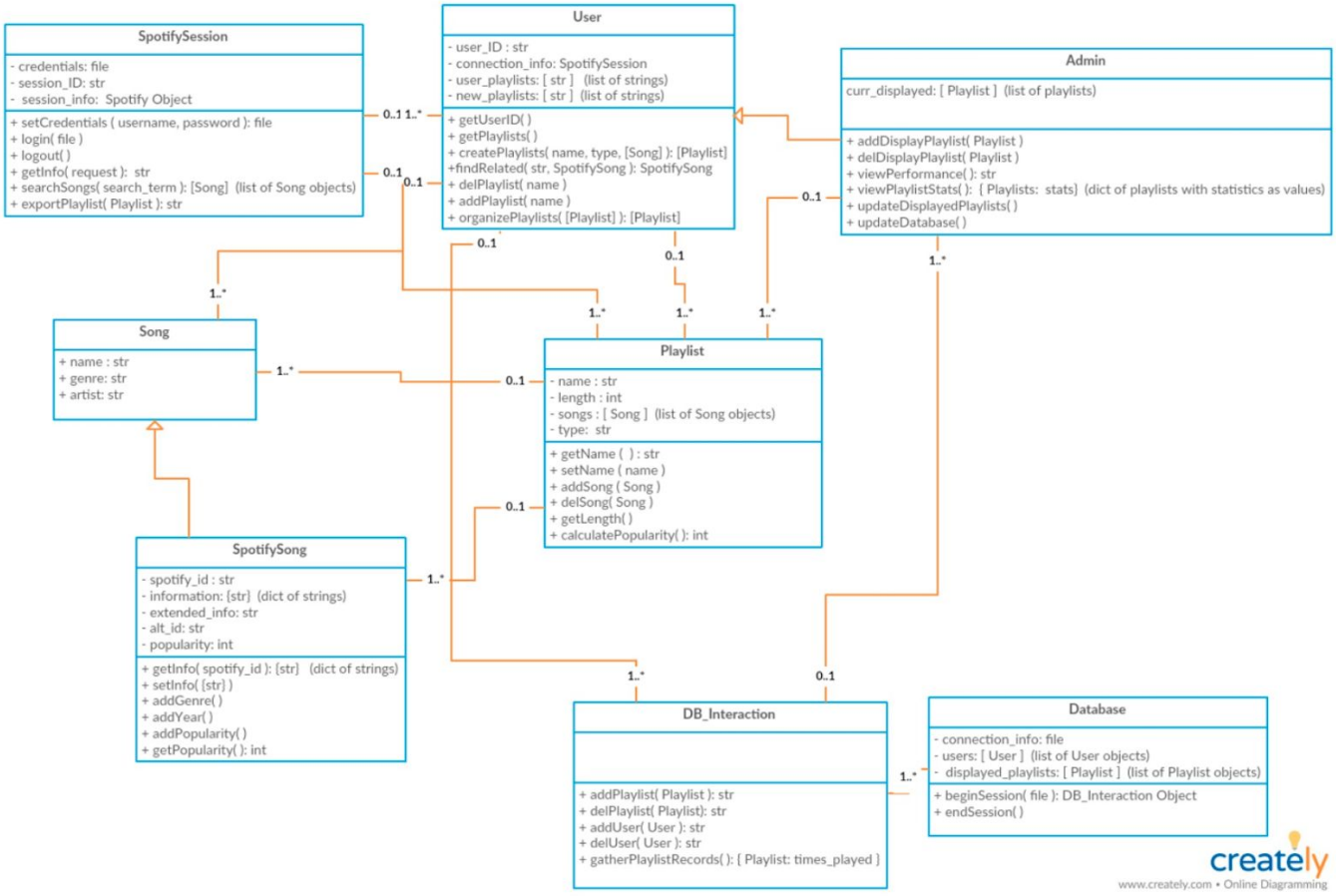
Table ID	Table Title	Features
UR - 02	User requirements	Users are able to add auto-generated playlists to their Spotify account
UR - 03	User requirements	Auto-generated playlist can based on year

UR -06	User requirements	Update popular playlists page based on what users are listening
UR - 11	User requirements	Users are able to login into the system
FR - 01	Functional requirements	System allows people login with their Spotify account
UC - 01	Add auto-generated playlists	Add auto-generated playlist into user’s spotify account
UC -02	List the popular playlists	Based on input, list most popular playlists

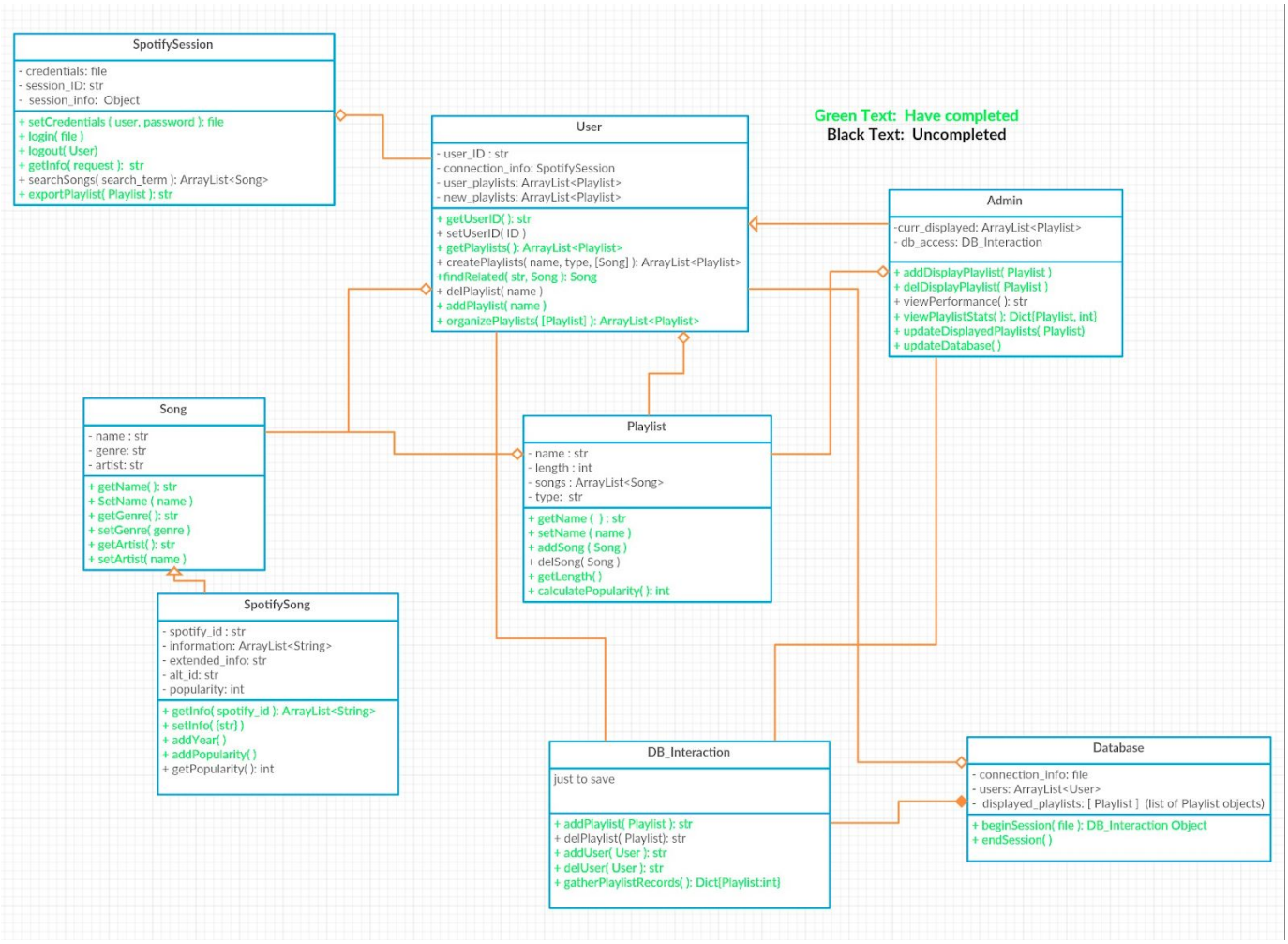
List the features were not implemented from Part 2 (table with ID and title)

Table ID	Table Title	Features
BR - 01	Business requirements	All actors need to have a special user name
UR - 01	Business requirements	Users need to be able to search for songs by name
UR - 07	User requirements	Users can view the popular music playlists

The class diagram from part two:



The final class diagram:

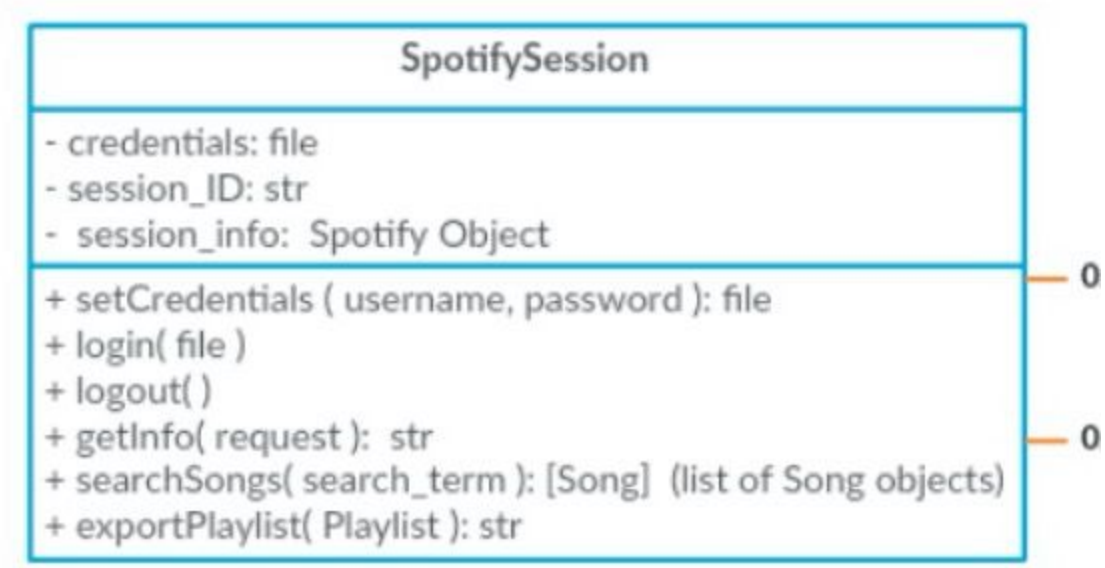


Discussion:

Our two class diagram don't change a lot. Setting up design first is a very efficient way for creating a project. When you are implementing functions for project, always look back to your class diagram that you created before

will make your logic more clear. This will definitely improve the quality of your project. Saving time is a reason from another perspective. Additionally, front design will always allow developer to check what have been done and what parts are still missing. In general, this is a great step to move for project developement.

Design Pattern: Singleton pattern



What we learned:

We believe that the most important thing from this class is learning a concept of how to maintain a code not just for now, but for the future. This can be achieved by using design patterns. We stepped through out project using singleton. After finishing this project, we know how important it is to having design part done before actually starts coding. This can help us a lot when we move forward on implementing functions. Once we have all of design of our project, we can always check which parts have already done, and which parts are still missing. Thus, this is a very high efficient way to start and implementing a project and this is also the