

Final Report

TinyTunes

Team Members: Sam Berger, Sam Busser, Jasmine Bascom

Final System State:

As our system stands now, it allows the user to sign in with their Spotify credentials, then give our app permission to modify their account and playlists. Then, the user is shown a list of their current playlists and is asked which playlist they would like to modify/prune. The app then takes the user to a new page showing what song has been selected to prune, shows the album cover of the song and allows the user to play a 30 second demo of it. The user is given a choice of deleting the song or not. If they choose to delete the song from the playlist, they are taken to a page confirming the song has been deleted, with a button to take them back to the home page. If they choose to not delete it, they are taken back to the list of their playlists.

List of features:

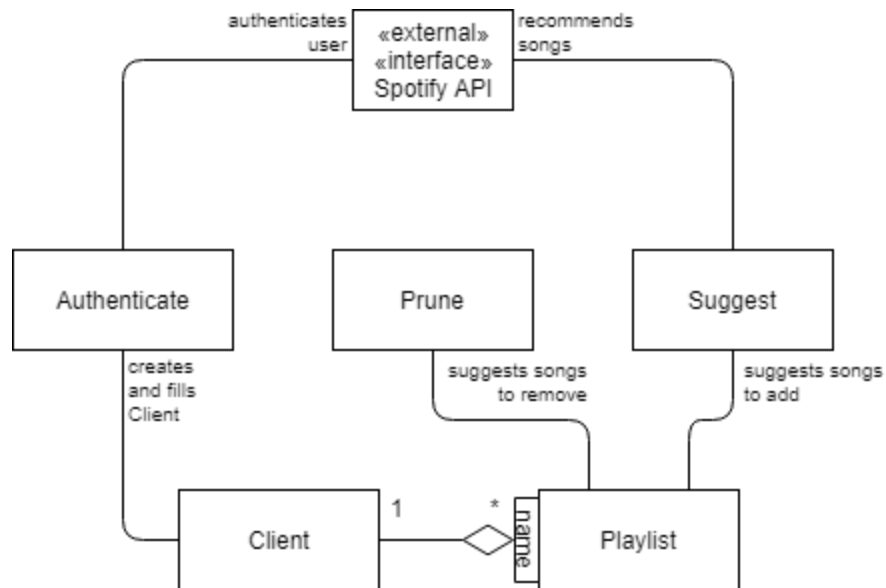
- Log in and authenticate with Spotify
- Display user's playlists
- Use algorithm to determine best song to remove from selected playlist
 - This is done by using Spotify's recommend a song feature. We call this API on the selected playlist, then use the Spotify API's in-depth song analysis to compare the recommended song to every song in the playlist. While doing this, we find the song from the playlist that has the highest numerical difference from the recommended song and choose that song as the one to prune.
- Display song that has been chosen along with album cover and 30 second sample
 - Also give user choice to remove or not remove song
- If remove button clicked, song is removed from user's playlist
- If don't remove button clicked, song is not removed from user's playlist and user taken back to playlists page.

Class Diagram:

Place holder

Final Project vs. Initial Design

Homework 4 Class Diagram:



Discussion of changes:

As can be seen from our two class diagrams, our project scope has changed quite a bit. In our original design, we did not consider the challenges that would be faced with hosting our live web application. This is why our original class diagram was pretty simple. The majority of our class diagram now is taken up by simply hosting a web application and dealing with users and permissions. The **Authenticate** class and the **Prune** class as seen in our original class diagram ended up becoming Django applications as a part of our web site. These applications are required to use our HTML code along with our Python logic for the site. We did create the **playlist** class, and we created a new class, **Song**, as well. One other major difference between our initial and final design is the fact that we do not have the suggest feature in our final design. This is simply because of a lack of time. We are very close and have most of the pieces in place for it, but we ran out of time to do a good job on it. This is something we would like to add in the future.

Third-Party Code vs. Original Code: