

BPM: A Better Playlist Builder

Alana Anderson (afa2132), Katie Pflieger (kjp2157), Matt Kronegold (mak2265), Tristan Orlofski (tio2001)

Part 0: Demo Details

We completed our demo on Monday 12/3 at 1pm. During our demo, we walked through the most recent iteration of our application which includes review mode, shuffling of the playlist creation album, name/save playlist in addition to the features that were previously implemented. Additionally, we reviewed our test coverage, which includes both unit and integration testing. We walked through the `test_generation.py` file in order to show that this is not a traditional unit test, but an integration test. The test checks that Spotify correctly interacts with a hardcoded account that has only a few songs in its library and that our generation playlist function outputs what is expected.

At the conclusion our demo, we discussed the next steps for our final iteration. We plan to implement song preferences, prompting the user if they would like to open spotify, and caching. Our cache will save the generated playlist information if the user is not able to save it to their Spotify account prior to exiting the application. This will allow the user to resume saving their playlist when they reopen the application without losing their history.

Part 1: User Stories

The specific user stories and conditions of satisfaction that were demonstrated are as follows:

1. Login:
 - a. We implemented a persistent authentication flow, where a user can create a BPM account and authenticate with Spotify
 - b. **[New]**: We added functionality to automatically refresh the user's access token, if expired
2. Welcome:
 - a. If the user already has a BPM account, they will be asked to enter their username
 - b. If the user does not have a BPM account, they will be asked to create one and authenticate with Spotify
3. Enter Inputs:
 - a. The user can specify length of playlist, start speed, and end speed and we will build a playlist that matches their specifications
 - b. If the user enters an input that is invalid (such as length of -10), they will be asked to try again
 - c. **[New]**: As discussed during our previous demo, we added the ability for a user to choose a start speed that is faster than their end speed
 - d. **[New]**: We also added shuffling to our playlist creation album, so that users who choose the same inputs won't always have the same credentials
4. **[New]** Review Mode:

- a. The user can choose to 0) swap a song, 1) re-do the playlist with different parameters, or 2) keep the playlist the way it is
 - b. If 0) user is asked which song they want to swap and taken back to the beginning of “Review Mode”
 - c. If 1) user is taken back to “Enter Inputs” to enter new parameters
 - d. If 2) user is taken to “Name Playlists”
5. **[New]** Name/Save Playlist:
 - a. The user is asked to provide a name for his/her playlist
 - b. The playlist is created with the given name and saved to the user’s Spotify account (so that if they open Spotify, they can view and listen to their playlist)
6. Logout:
 - a. The user can type “logout” at any point and be logged out of the system
 - b. The next time the program is run, the user can either login again with their BPM account (so that they do not have to re-authenticate with Spotify) or let another user login

Part 2: CI Mechanisms

For our project, we are using Travis CI hosted in the cloud (travis-ci.org). The CI is configured to run static code analysis using pylint, testing with pytest, and coverage reporting with pytest-cov everytime a contributor pushes to the master branch. The results of both tests are pushed to our GitHub repository on a separate branch. By pushing to a separate branch, we prevent an endless loop of the CI being triggered by its own pushes.

Part 3: GitHub Repository

Repo: <https://github.com/mattkronengold/bpm>

Pre-commit CI: <https://github.com/mattkronengold/bpm/blob/master/pre-commit.sh>

Post-commit CI: <https://github.com/mattkronengold/bpm/blob/master/.travis.yml>

CI reports: https://github.com/mattkronengold/bpm/tree/travis_ci_output