BPM: A Better Playlist Builder

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

Part 0: User Stories

Title	**Actor**	**Description**	**Flow**
Login/Welcome	An active runner who loves to run to the beat of the music and wants to build a playlist for an upcoming run.	As a user, I want to be able to enter my login credentials one time so that I will not have to re-login next time I open the application (unless I've logged out).	<u>Basic Flow (new user):</u> ● User is welcomed to BPM ● User is asked if they have a BPM account (indicates no) ● User is asked if they have a Spotify account (indicates yes) ● User is asked to choose a username for their BPM account ● User is taken to Spotify OAuth page and copies the authentication key <u>Alternate Flow (existing user):</u> ● User is welcomed to BPM ● User is asked if they have a BPM account (indicates yes) ● User is asked to enter their BPM account ● User is logged in
Enter Inputs	An active runner who loves to run to the beat of the music and wants to build a playlist for an upcoming run.	As a user, I want to enter the length of my playlist, genre of music, start speed of my run (in steps per minute), and end speed of my run (in steps per minute) to be used for my custom playlist.	<u>Basic Flow:</u> ● User is asked to choose from a menu of genres for their playlist ● User is asked to enter the duration of their playlist (in minutes) ● User is asked to enter the start speed of their run (in steps per minute)

			● User is asked to enter the end speed of their run (in steps per minute)
Review Mode	An active runner who loves to run to the beat of the music and wants to build a playlist for an upcoming run.	As a user, I want to be able to review the details of my playlist and make any changes so that my playlist is complete before I begin my workout.	Basic Flow: ● User is shown their generated playlist ● User is asked if they want to 0) swap a song, 1) re-do the playlist with different parameters, or 2) keep the playlist the way it is ● If 0) user is asked which song they want to swap and taken back to the beginning of "Review Mode" ● If 1) user is taken back to "Enter Inputs" to enter new parameters ● If 2) user is taken to "Name Playlists"
Name Playlist	An active runner who loves to run to the beat of the music and wants to build a playlist for an upcoming run.	As a user, I want to be able to give a unique name to my playlist so that I can play it in Spotify.	Basic Flow: ● User is asked to enter a name for his/her playlist Alternate Flow (playlist name is already being used): ● User is asked to enter a name for his/her playlist ● An integer is appended to the user's input
Logout	An active runner who loves to run to the beat of the music and wants to build a playlist for an upcoming run.	As a user, I want to have the option to logout of my Spotify account from the command line app and allow a different user to login.	Basic Flow: ● User enters "logout" at any point in the application where input is accepted ● Next time I login, I am asked to enter my BPM username

Part 1: Acceptance Testing

Test Case	Boundary Condition	Valid Equivalence Classes	Invalid Equivalence Classes	Test Location
Length	An input that is an integer < 5, an integer > 90, a non-integer type, or NULL is considered invalid and logged.	Integer [5, 90]	Integer < 5 Integer > 90 Non-integer {i.e. “cat”, “$13”} NULL	https://github.com/mattkronengold/bpm/blob/master/src/test_inputs.py
Genre	For a list of genres indexed 0 through 8, every input that is an integer < 0, an integer >= 7, a non-integer type, or NULL is considered invalid and logged.	Integer [0, 8)	Integer < 0 Integer >= 7 Non-integer {i.e. “cat”, “$13”} NULL	https://github.com/mattkronengold/bpm/blob/master/src/test_inputs.py
Speed	An input that is an integer < 50, an integer > 300, a non-integer type, or NULL is considered invalid and logged.	Integer [50, 300]	Integer < 50 Integer > 300 Non-integer {i.e. “cat”, “$13”} NULL	https://github.com/mattkronengold/bpm/blob/master/src/test_inputs.py
Logout	An input of “logout” should prompt the system to log the user out, as specified in the user story.	“logout”	“log out” “ogout” “logou” Integer {i.e. 5, 10} NULL	https://github.com/mattkronengold/bpm/blob/master/src/test_logout.py

To measure the coverage of our test suite, we added a coverage tool to our post-commit CI with reports sent to https://github.com/mattkronengold/bpm/tree/travis_ci_output/. We choose to use pytest-cov, a tool that monitors the program, noting which parts of the code have been executed and analyzing the source to identify code that could have been executed but was not. As expected, some of our python files show low coverage because they interact directly with a user's spotify library and are therefore unable to be tested with unit testing. Instead, we added a simple regression test for generation.py. This regression test also utilizes some code from auth.py. Bpm.py has 0% coverage as it is just the main driver function of the

application. Database.py is tested completely (in test_databases.py), but this is integration rather than unit testing, and thus there are no equivalence classes to report.

Part 2: 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