Riffle – Spotify Listening Insights
Team Name: Riffle
Members: Braeden Watkins, Chris Ramos, Tyler Cartier

Introduction

Overview

Riffle is a cross-platform application that uses the publicly available Spotify API to provide users with useful statistics about their listening habits. The project aims to enhance user engagement with their music data, while also serving as a learning opportunity for the development team in API integration and React Native development.

Motivation

The primary motivation behind this project is to familiarize ourselves with APIs and learn the basics of React Native. Our overall motivation is to learn how to use common systems used in industry while still creating something that could be useful to our daily lives as well as others. All of the team members enjoy listening to music and wanted to develop something that could make our listening efforts more efficient. Specifically, we wanted to find a way to adapt our playlists to better fit our tastes, using statistics to do such.

Market Context

Riffle is based on the very popular Stats.fm app. Stats.fm also displays the user's statistics, however, they limit what can be seen and are missing some features that we at Riffle would like to implement. For example, neither Spotify nor Stats.fm keeps track of the number of times a user has skipped a song. We think this can be a very useful statistic that the user can use to think about their listening habits and potentially edit their playlists.

Team Background

None of the team members have experience with using APIs, React Native, databases, or the Expo framework. While we do not have direct experience with the systems we plan on using, Chris and Braeden both have experience working on larger-scale projects as they have both worked on a Unity game prior to this class. This is Tyler's first time working on a larger-scale project.

Customer Value

Customer Need

Riffle is not an app directed at the everyday Spotify user. Not everyone who listens to music cares about how many times they've listened to a song. However, there is a group of people,

we'll call music connoisseurs, that would like to see all of these statistics. These music connoisseurs aim to further their knowledge about their music listening habits to better hone their understanding of their tastes. Just using Spotify by itself does not provide the user with everything they might want to know. As such, we have decided to create this app.

Proposed Solution

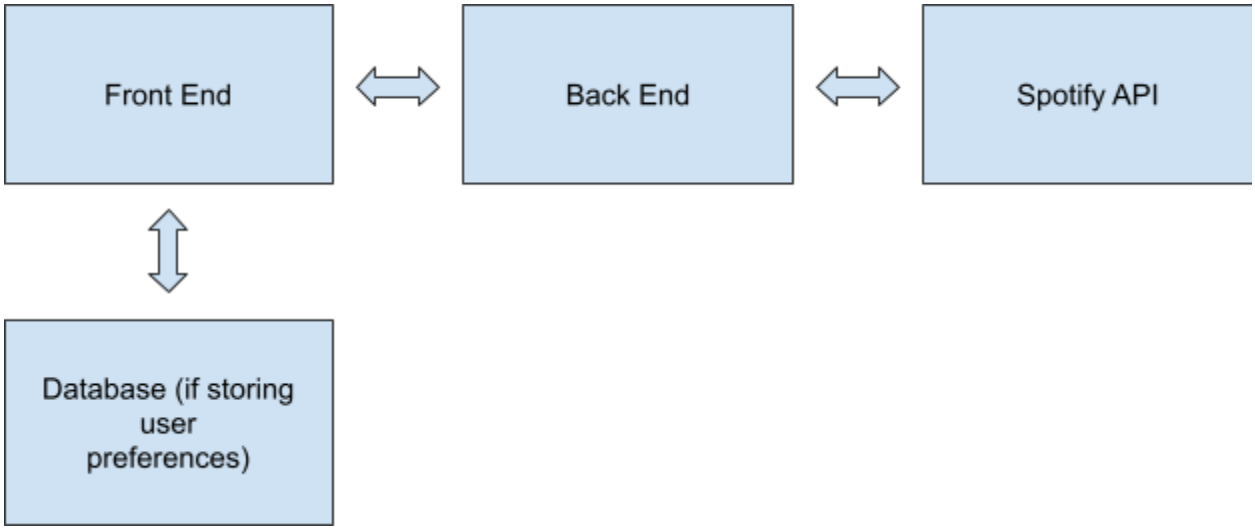
Our software will provide the user with an easy-to-use application that will track their stats without the user having to keep track of these things themselves. Currently, if someone wanted to know how many times they have skipped a particular song, they would have to keep track of this information themselves. For example, they would have to create some sort of manual tally and increment that tally by hand every time they skip a song. This is obviously tedious and not something anyone would want to do, so an application that does this for you could prove to make life easier and more efficient. This feature is something that has been mentioned to several friends of the developers and all had a keen interest in the idea. In addition, Spotify has a "Live Ideas" community section where people can propose ideas to implement into the application, and many users have agreed with the feature to track skipped songs but it has had no implementation.

Measures of Success

We will periodically ask our testers if using the app has had a positive effect on their listening experience. Hopefully, once testing is complete and the testers are satisfied with the product, we will be able to push the application into production. This application is not necessarily something we expect a multitude of people to use, but if even a few people that we do not personally know used the application, we believe that would be a great measure of success. If we do end up having consistent users, we could track how often those users are visiting the application as a metric to determine how useful the app is.

Technology

Our application will use Spotify's API to access user data and configure that data into easy-to-see listening statistics. The main components of our system would be the front end, which provides a clean, easy-to-use UI, and the back end, which connects the application to the Spotify API.



A minimal system that would have value to the customer would be an app that allows users to log in with their Spotify account and view basic statistics about their listening habits. Any additional features that users of Spotify have suggested that have not been implemented that we are able to add would be ideal. The more we can forge our application into something that appeals to what users of Spotify want, the more our customers will value our product.

We will have beta testers who will be provided with the application after each major milestone to make sure everything is working. We will utilize the already implemented testing feature within the application we created on Spotify's developer website. In order to bring our project to life, we will utilize the tools of React Native, Node.js, Expo Framework, Spotify Web API, and potentially a database.

Team

No one on the team has built something like this before. It will be a new learning experience for everyone, but we think it is manageable. The tools being used are also new to the team; however, some of them are very similar to others that have been used. The specific roles that team members will rotate through are UI, API calls, and database production. As mentioned, these roles will rotate so that each developer can gain experience with each role.

Project Management

We believe that it is entirely possible to have a working application by the deadline. We plan to meet face-to-face on Tuesdays and Thursdays after class, as well as on Discord. We have already created a server for our application where we can share screens and talk about the progress that we have made.

- Week 1 (February 17th)
    - Successfully grab/store authentication tokens in order to make Spotify API calls
- Week 2 (February 24th)
    - Make API calls/Begin UI (frontend) development
- Week 3 (March 3rd)
    - Continue UI development (make it look pretty)
- Week 4 (March 10)
    - Work on storing stats made available from Spotify API
- Week 5 (March 17)
    - Continue working on storing stats/start developing the skip feature
- Week 6 (March 24)
    - Continue working on the skip feature (our main component)
- Week 7 (March 31)
    - Database implementation
- Week 8 (April 7th)
    - Continue developing the database and add the finishing touches
- April 14th & 15th
    - Change or update UI if needed, debug features

Constraints

Our app must comply with Spotify's API Terms of Use, ensuring proper authentication, attribution, and restrictions on data storage. If our application goes out of the scope of the classroom, then eventually, we'll have to adhere to relevant privacy laws. We will also avoid Spotify's copyrighted content to prevent property violations. The primary ethical concerns involve user privacy and data transparency. Allowing users to delete their data and informing users of which data will be collected would address these. All of our data will be coming from the Spotify API, which provides access to a user's listening history, top tracks, and playback data. We will not be collecting data manually.

Ideally, we implement many tracking statistics, however, if we need to descope, we at the least think that tracking the skipped songs would be good to implement. Our application would still be useful even if we have to narrow it down.