

Initial Architecture Overview

Scorify

Team 25

Blake Carlson, Jack Bauer, Nifemi Lawal, Dellie Wright, Logan Smith

Project Synopsis

A web app connecting users' Spotify accounts to analyze listening data, compute genre diversity and taste scores, and display personalized dashboards, leaderboards, and daily recommendations.

Architecture Description

Overview

Scorify consumes users' Spotify listening history to visualize personal music trends, display data-driven insights with graphs, and calculate unique metrics such as genre diversity and taste score. It transforms raw listening data into an interactive experience where users can explore their listening habits and compare them with others through a leaderboard.

Data Path

Scorify uses a PostgreSQL database to store user data once they link their Spotify account to Scorify. After authentication, Scorify securely retrieves and stores each user's listening history and profile information. By persisting this data, the system can reference its own database instead of repeatedly calling Spotify's API. This stored data also powers Scorify's global leaderboard, which updates whenever users log in. Each login triggers a refreshment in that user's data, ensuring accuracy while maintaining full compliance with Spotify's data access policies.

System Architecture

The system centers around a clean, data-driven interface built to make user insights immediately understandable. At its core, it is a personalized dashboard that visualizes listening behavior through a viewable history and summaries such as a taste rating and

genre diversity score. To the right of the dashboard, a dynamic leaderboard ranks users based on either genre diversity or taste score. Users can seamlessly toggle between these leaderboard views to compare their listening trends within the Scorify community. This forms the basis for the client portion of the web application. The server portion of the web application is responsible for handling all communication with the PostgreSQL database and the Spotify API servers.

Metric Computation

Diversity Rating – Computed using a normalized form of Shannon’s Entropy, this metric captures how evenly a user’s listening time is distributed across different music genres. The more evenly distributed the listening activity, the higher the diversity score. Scorify retrieves a user’s listening history and aggregates the number of tracks per genre from the local PostgreSQL database. These counts are converted into probabilities that represent the proportion of listening activity per genre.

Let p_i denote the probability that a song listened to by the user belongs to genre i , and let N represent the total number of distinct genres the user has listened to during the sampled period. The diversity score D is calculated as:

$$D = - \frac{\sum_{i=1}^N p_i \log_2(p_i)}{\log_2(N)}$$

This formula normalizes the entropy value, ensuring that D always falls between 0 and 1. A score near 1 indicates a highly diverse listener who engages with many genres evenly, while a score near 0 indicates a narrow listening focused on a single genre.

Taste Score – A comparative metric that evaluates how closely a user’s listening diversity aligns with the Scorify development team’s average diversity. It reflects how similar a user’s listening habits are to a precomputed baseline.

Let D_u be the user’s diversity rating, and D_{dev} represents the average diversity rating of the development team. The taste score T is calculated as:

$$T = 1 - |D_u - D_{dev}|$$

Users whose listening diversity closely matches the team's average receive a score closer to 1, while those who differ significantly receive a score closer to 0.

Security and Data Privacy

Scorify implements multiple layers of security to protect user data. Authentication and data access are managed through Spotify's official OAuth 2.0 Authorization Code Flow, ensuring that all user data is retrieved securely and only with explicit consent. When a user chooses to link to their Spotify account, they are redirected to a Spotify authorization page where they can review and approve the requested permissions. After the user grants access, Spotify returns a short-lived authorization code that Scorify exchanges for an access token and a refresh token. The access token allows temporary access to the Spotify Web API, while the refresh token enables Scorify to obtain new access tokens without requiring the user to log in again. This design minimizes the need for users to repeatedly enter their credentials and ensures that access to Spotify data remains secure and user controlled.

Program Data Flow Diagram

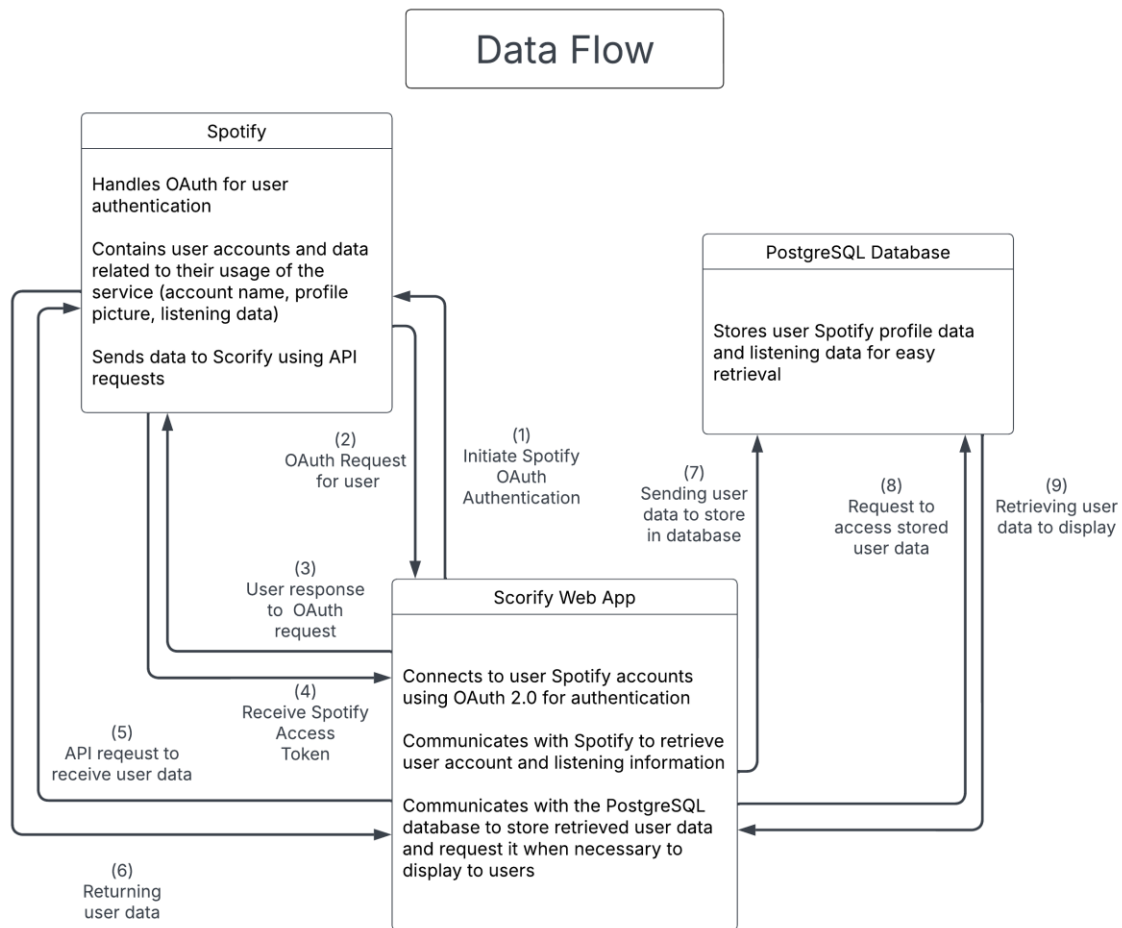


Figure 1

The three main entities that handle and process data are Spotify, Scorify, and a database running the PostgreSQL database engine. These together allow Scorify to retrieve user's Spotify account data and store it for easy retrieval. The data flow for a standard use case is numbered (1) through (9). The procedure begins with a user logging into their Spotify account using an OAuth request. Once successful, the Scorify web app can request specific user data such as their account profile picture from Spotify. It can send this data to the database for easy retrieval, preventing multiple API requests for the same data. Scorify can query the database for any needed stored data or use an API request for any new data.

Client-Server Interaction Overview

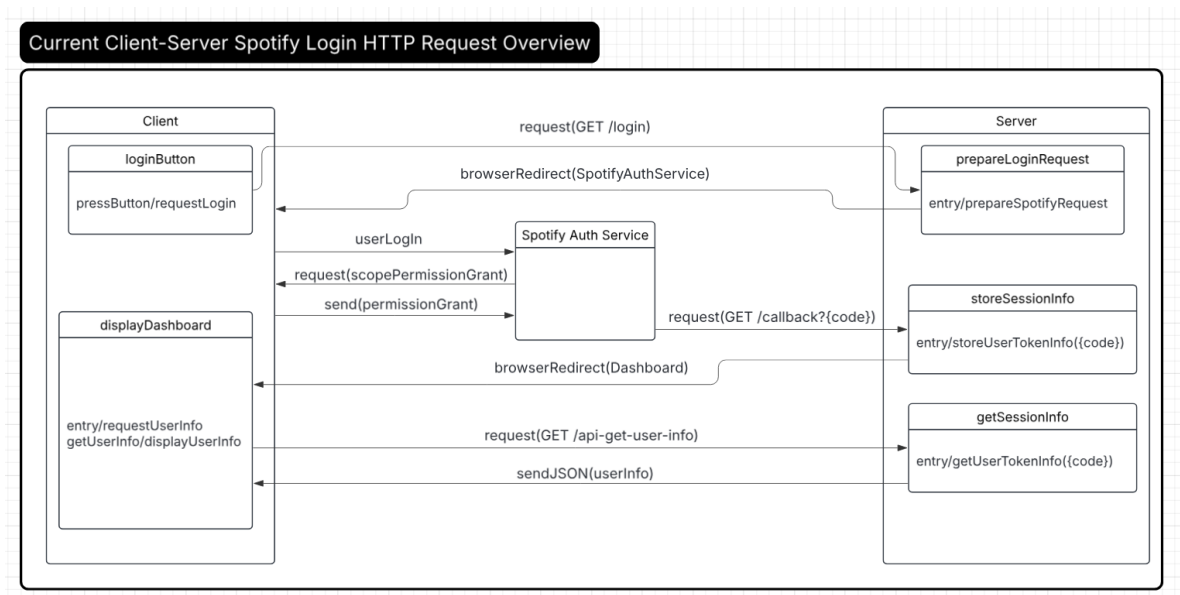


Figure 2

The Scorify web application serves as a mediator between the user, the PostgreSQL database, and Spotify. However, this web application is divided into two parts. The client, which is run in the browser and is responsible for displaying data to the user and the server, handles all interactions with external hosts such as the Spotify API servers and the Scorify database. The above diagram shows the general communication flow between the client and the server during the user login process. It particularly focuses on the specific HTTP requests needed.

UI Navigation Diagram

Figure 3 shows how an end user would navigate Scorify. First, the user reaches the Scorify landing page and begins logging in with their Spotify account. After a successful login, the user waits on an intermediary loading page while their Spotify data is retrieved. Then the user reaches their dashboard displaying information to them about their listening history, taste score, and diversity score. Clicking on another user's profile will let you view it. There is also an option to click on the "About" page which explains the formulas for the score calculations. Clicking on your profile will navigate to your dashboard if you are not presently viewing it.

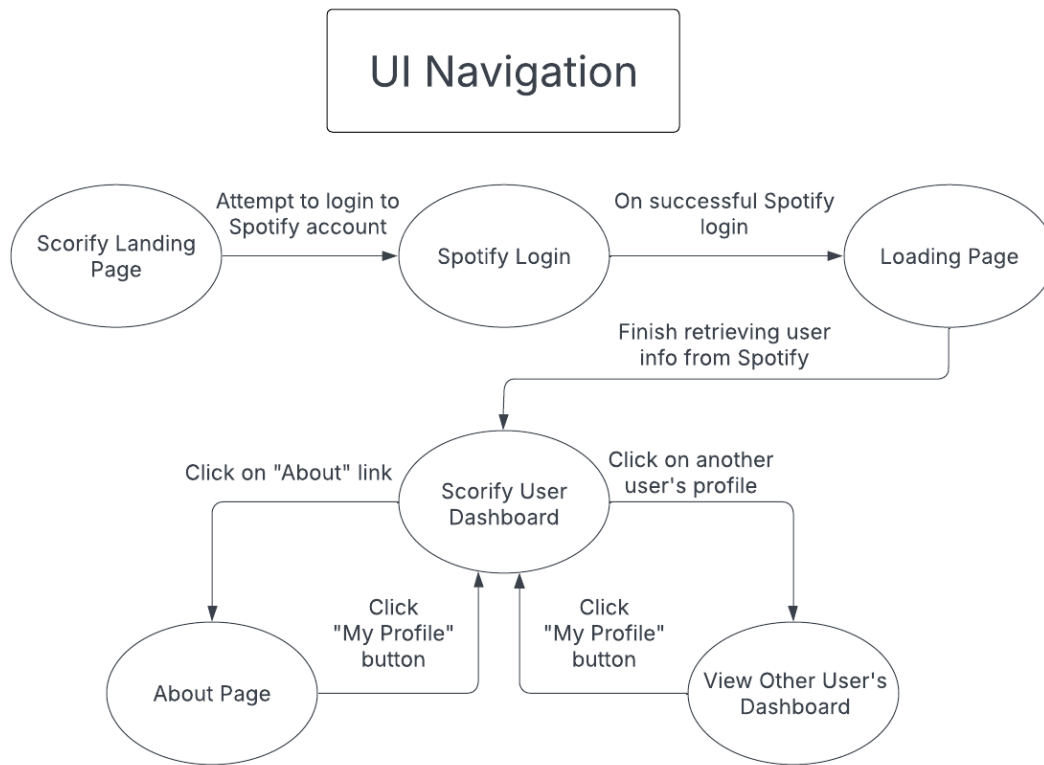


Figure 3