Sprint 2 Requirements Artifacts

Team 25

Requirement #5: Fetch user listening history data from Spotify (3)

* Detailed description:
  + When a user successfully logs in via Spotify, fetch their user listening history so it can be used in calculating their music scores.
* Planned implementation
  + Have the JavaScript client send an API request to Spotify on a successful login, so their data is automatically retrieved.

Requirement #7: Establish connection to cloud database on app launch (8)

* Detailed description:
  + When our application is launched, we will establish a connection with the Scorify external database. This is needed to store important information and limit the amount of Spotify API requests the application makes.
* Planned implementation:
  + Near when the application is launched, at a useful time, such as when the application makes a request to load the logged-in user’s data from Spotify, the application will also establish a connection with our external database and begin loading information for other users. This will allow us to access the information needed to display the leaderboard and the content needed for other users’ dashboards.

Requirement #8: On Spotify connection failure, display error message (3)

* Detailed description:
  + When a user attempts to log in via Spotify, but their credentials are invalid, provide a simple message for redirection to the homepage or to retry logins.
* Planned implementation:
  + Create a new logical flow for denied authentication in JavaScript client. Ensure that redirections exist for target locations after being denied authentication.

Requirement #9: On database connection failure, display error message (3)

* Detailed description:
  + When the web application attempts to connect to the external database and the connection fails, display an error message to the user so that they know something went wrong.
* Planned implementation:
  + Display a specific error message detailing what went wrong on the webpage to the user. This could be a popup message or a message that replaces content on the screen.

Requirement #13: Store user's username in cloud database (3)

* Detailed description:
  + Upon authentication, ensure the user’s Spotify username is a key to reference a user’s Spotify data in our Database. This username will be uploaded to our database and stored for later retrieval.
* Planned implementation:
  + Create a new table for users within our database, ensuring fields for username, and other Spotify data such as listening history and profile picture, and any other required relations exist.

Requirement #14: Store user profile picture in cloud database (3)

* Detailed description:
  + When the web application retrieves the user’s profile picture from the Spotify API, we then provide this information to the Scorify database for storage and later retrieval.
* Planned implementation:
  + On fetch of a user profile picture, upload a copy of it to the database tied to that specific user through their Spotify username.

Requirement #15: Store user listening history data in cloud database (3)

* Detailed description:
  + Similar to the username and user profile picture, the user’s listening history, as retrieved by Spotify, will be stored in the database and tied to that user for later retrieval.
* Planned implementation:
  + On receipt of this data from Spotify’s API, we will format this data and upload it to our database and tie it to the user using their username as the primary key.

Requirement #18: Create user dashboard layout and page to display relevant data (5)

* Detailed description:
  + Create a dashboard that collects the user’s relevant data – song tastes, diversity score, listening history, etc. - and displays this data in an easily navigable format, available upon login to our service.
* Planned implementation:
  + Create basic dashboard, connected to our back-end database.
  + Define front-end functions to render each user’s data accordingly.
  + At a minimum, it will display the user’s username, profile picture, listening history, diversity score, and music taste rating.

Requirement #21:. Display user's listening history on user's dashboard (3)

* Detailed description:
  + After we retrieve the user’s listening history from the database, we will display it to the user on their dashboard in an easily readable and understandable way. We may limit the number of songs to a certain constant value, as displaying the entire history may be difficult.
* Planned implementation:
  + This will be displayed on the user’s dashboard as a list-like structure that is either scrollable or navigated through with pages. Some reasons to limit the number of songs displayed are clarity, the normalization of data across users, and avoiding excess Spotify API requests.

Requirement #32: Add an about page with useful data for the user (3)

* Detailed description:
  + There will be a specific page dedicated to providing information about our product that may be of interest to the user, such as how their scores are calculated. It will be easily accessible from our web application.
* Planned implementation:
  + On the dashboard, there will be a button the user can click to access the about page. It will contain relevant images and paragraphs of written content in an effort to help the user better understand our application.

Requirement #33: Explain diversity score formula on about page (1)

* Detailed description:
  + The about page (which will be accessible from the dashboard), will explain the formula used to calculate the diversity score. This should help the user have a better understanding of what the score actually means.
* Planned implementation:
  + We will host a paragraph of content on the about page explaining the score as detailed above as well as showing the formula that is used.

Requirement #34: Explain music taste rating formula on about page (1)

* Detailed description:
  + On the about page, which will host a collection of information that may be of interest to the user, we will also explain the music taste rating and talk about how the formula was derived and what the score means to the user, so they can have a better understanding of the purpose of this score.
* Planned implementation:
  + We will host a paragraph of content on the about page explaining the score as detailed above as well as showing the formula that is used.