



Track Records

Senior Design

Gustavo Chavez, Patrick Arzoumanian, Abraham Mookhoek, Spencer Whitehead, Ahmed Ullah



Department of Computer
Science and Engineering

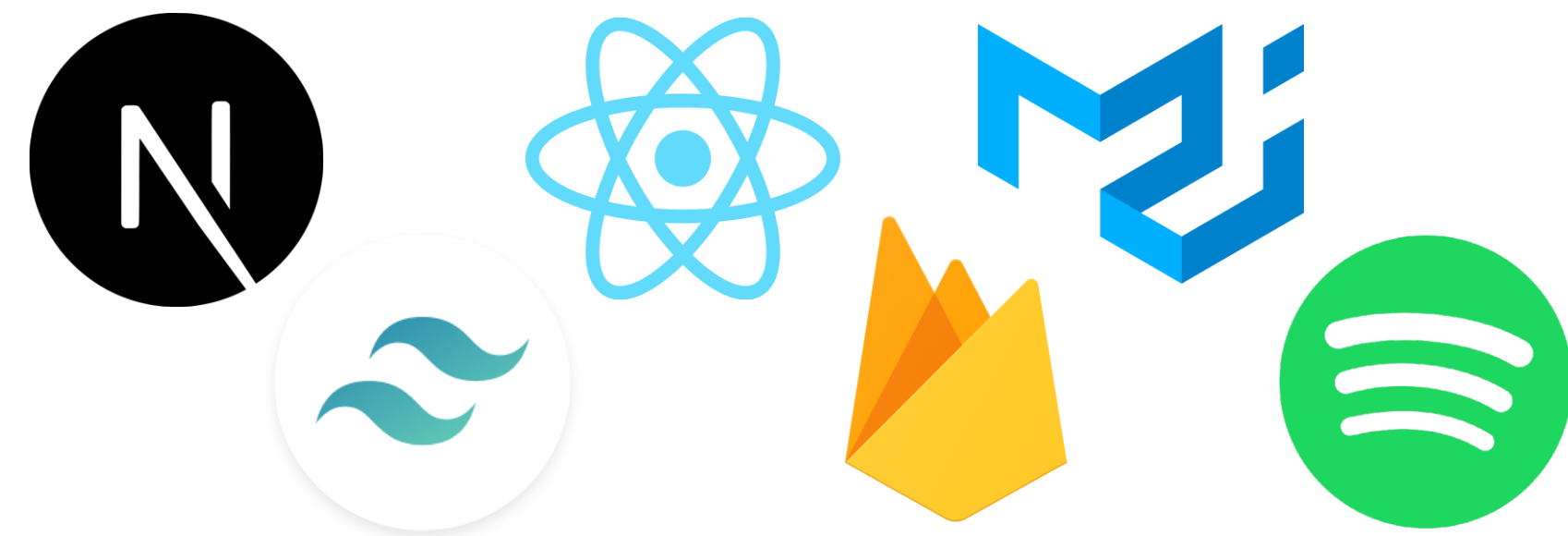


Background

The concept of Track Records, a music analytics web application, is to offer a variety of data analytics tools in an easy and intuitive form. Users will be able to utilize these tools to look back on their Spotify listening habits. The data in question relates to each user's provided music streaming history. Users will be able to understand their listening habits by utilizing our application's data visualization as well as keep a detailed journal of their musical history.

To make this possible, some key requirements include a calendar interface to visualize the user's listening history, a statistics interface to provide another visual representation of other fringe statistics, and a journal interface to provide users a way to document their thoughts on a given day of listening to a specific song.

To achieve these requirements, the scope of our project will be primarily based in popular web-development technologies such as Firebase databases and next.js. Since our web-app is user focused, we will also be handling the authentication of their Spotify accounts as well as a database to store their data.



Key Requirements

- **Login/Registration Page** : If not logged-in, when users open the web-page, they will be met with a login/registration screen. This will simply prompt the user to either create a new account or to login with an existing one. An account is essential since our servers will need to store the listening history of each individual user.
- **Calendar Interface** : Upon login, the user will be met with a calendar view interface. This view will show users information about when songs were added to playlists, which dates have journal entries, and more. The calendar will show the current month by default.
- **Journal Interface** : The journal interface will present users with a list of entries already added to the journal as well as a button that lets users create a new entry. When viewing an individual entry, the user may make edits or return to the entry list.
- **Statistics Interface** : The statistics interface will present users with a set of drop-down menus used to determine graph-type, data metric, as well as individual data entries. Once all of the parameters have been selected, a visual representation of the selected data will be presented to users.
- **Settings Interface** : The settings menu will show users a series of editable options in relation to the user's account, linked streaming services, the data stored on the product's servers, and more.

Architectural Design

Track Records will be structured with three layers, Client-Side, Business, and Data. Each of these layers combine to create a cohesive system capable of providing an intuitive and responsive experience for the user.

The user interface of a website that allows users to communicate with our web server via a browser is known as the Client-Side Layer. In order to show the data that the client has requested, the client-side talks with the server to send HTTP requests and receive HTTP responses to display the requested feature back to the user. Our web-app has many sub-components such as the login view, calendar view, journal view, and the settings view.

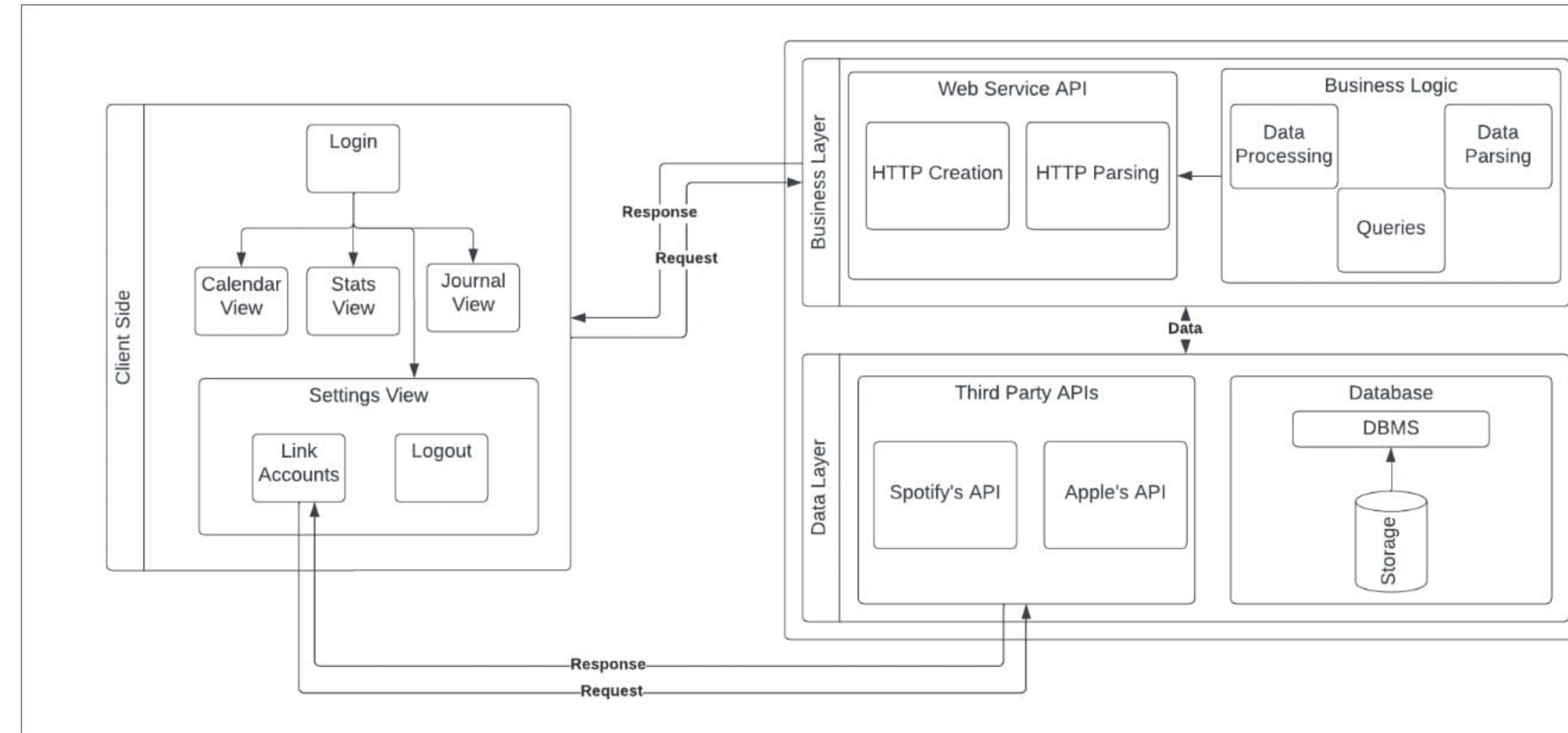


Figure 1. Track Records System Architecture

The business logic of the web application is managed by our Business Layer. It is made up of models, controllers, and services that carry out the functions for user requests. To retrieve or modify data as needed, the business layer communicates with our Data layer. It also houses our web service API which is primarily responsible for communicating back and forth between our Client-Side and Data Layer.

Our Data Layer houses the storing, querying, and requesting of data from either 3rd party APIs or the database service itself. While the 3rd party APIs are queried via our database connection housed in our back-end, our data is stored securely server-side. This layer can also house our web-app's project files.

Implementation Details

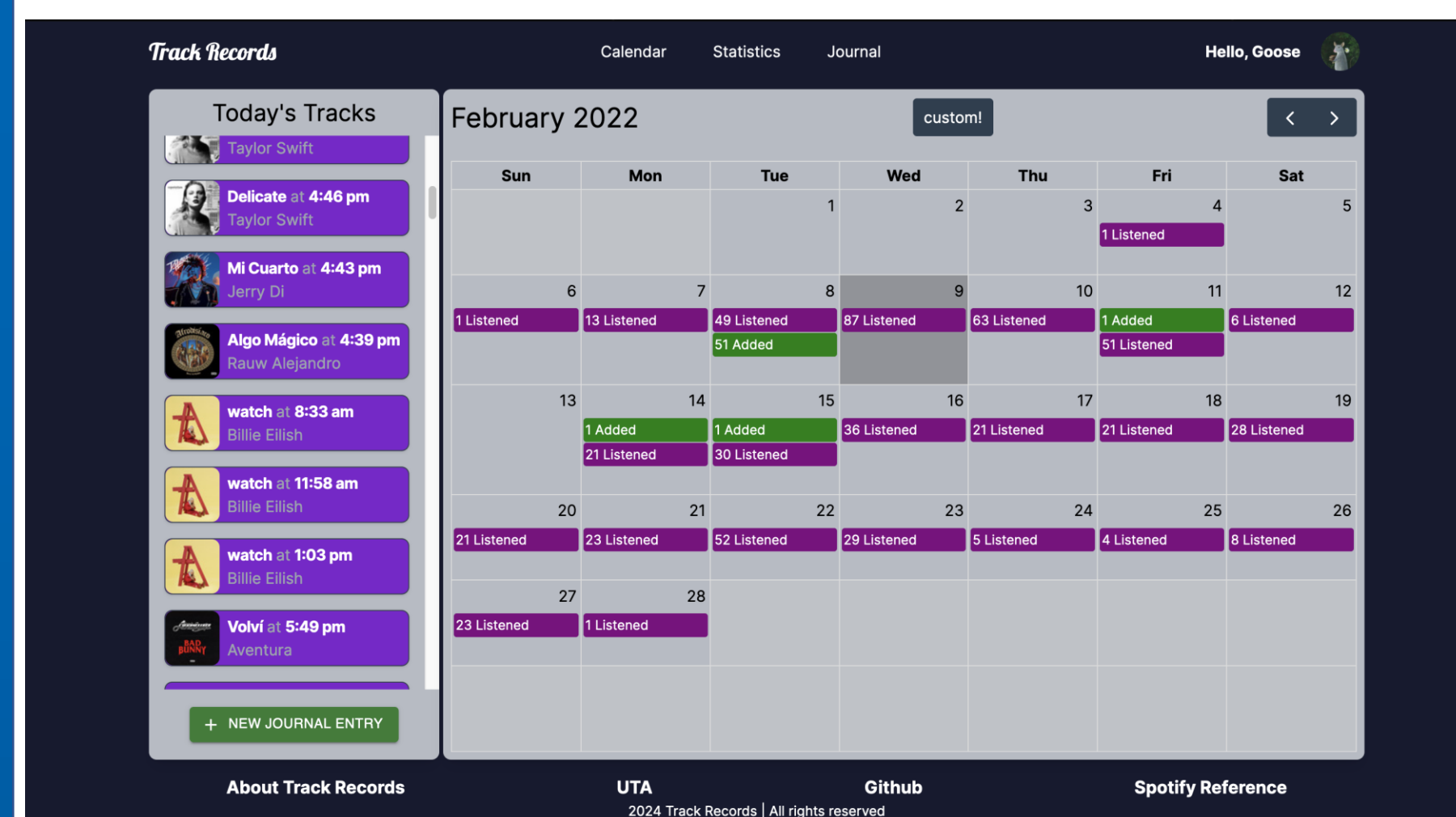


Figure 2: Calendar View Interface

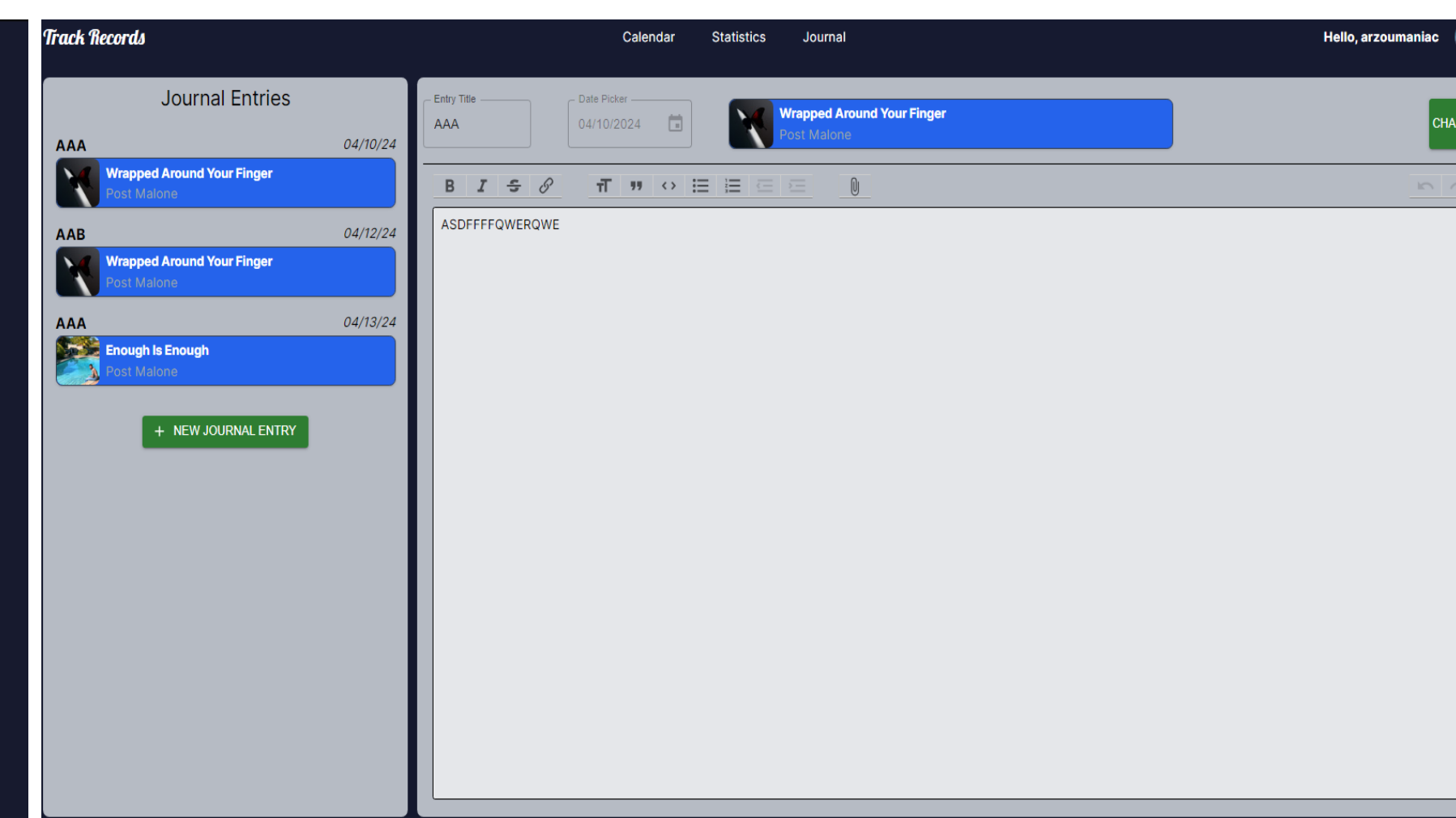


Figure 3: Journal View Interface

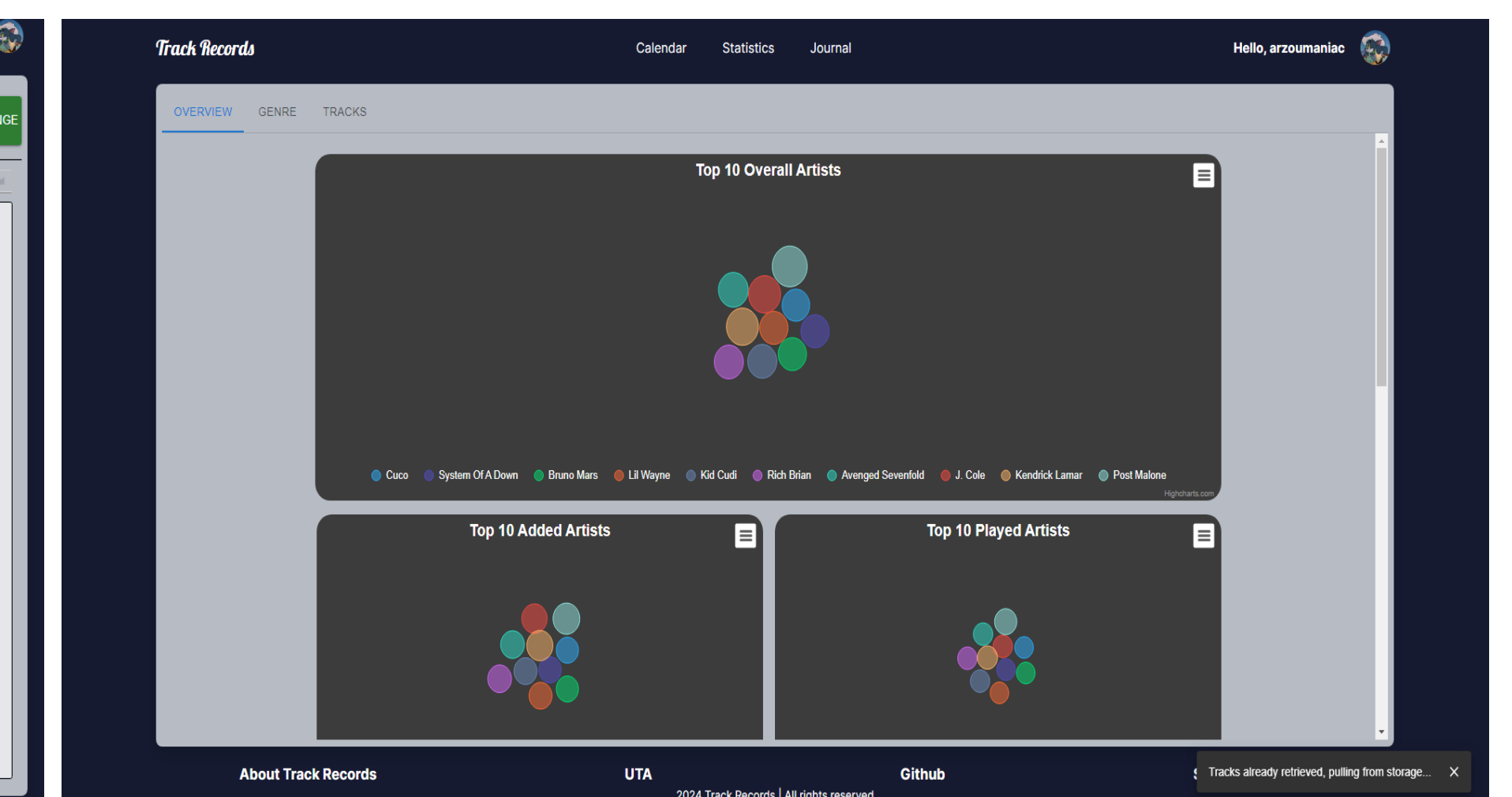


Figure 4: Statistics View Interface

Conclusions and Future Work

- **Conclusions:** Track Records has successfully implemented the essential interfaces— Calendar View, Journal View, and Statistics View— enhancing user engagement and understanding of their music listening patterns through detailed analytics.
- **Future Work:** Future enhancements will include integrating with Apple Music to extend to our user base improve the app's functionality, offering a more comprehensive music analytics experience.

References

1. G. V. R. Staff, "Music streaming market size & share analysis report, 2030," Jul 2023.
2. C. Eggertsen, "How does spotify wrapped work?" Dec 2021.
3. LastFM, "Explore top music powered by your scrobbles," 2023.
4. S. Staff, "Stats.fm," 2020.
5. A. Olivero and N. Patel, "Obscurify music."
6. A. Aufrichtig and E. Lee, "Discover quickly," 2019.