YEAR-END CHARTS
Billboard Top 100 Songs

By: Pranav Chakilam & Mark Soria

Purpose

Provide Users with Song Suggestions based on their listening history

- Collect data of liked and disliked songs
- Suggest similar songs according to the collected data from User
- Predict future top 100 songs based on previous data

System Description

Integrated Development Environment (IDE):

- Visual Studio Code (VS Code)

Database:

- SQLite3

Language:

- FrontEnd: HTML, JavaScript
- BackEnd: JavaScript

Framework:

- React/Flask

Use-Cases

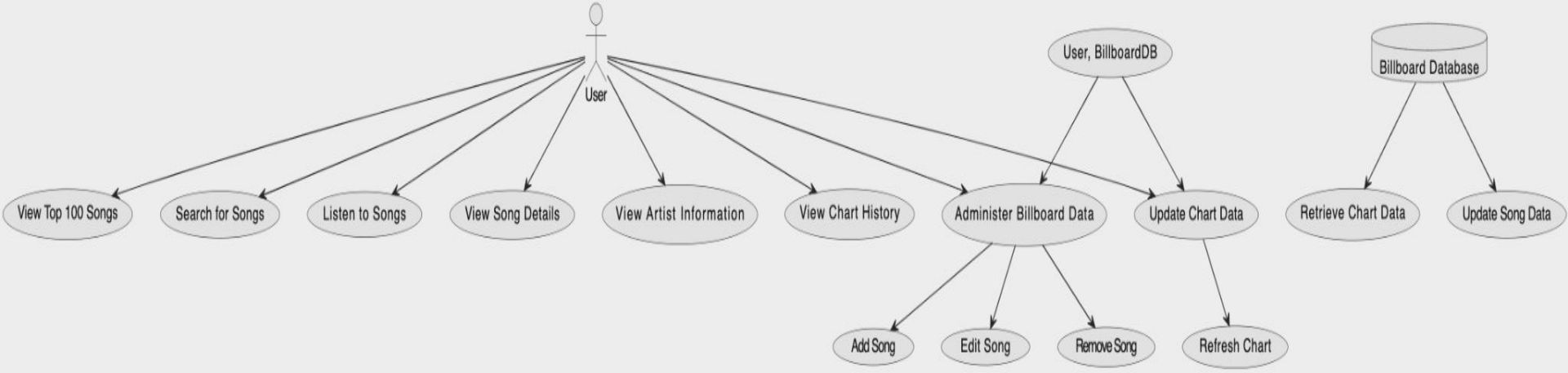
- **User Use cases**
  - View Top 100 Songs
  - Search for Songs
  - Listen to Songs
  - View Song Details
  - View Artist Information
  - View Chart History
- **Relationships**
  - Login → Authentication
  - Search for Songs → Song Details

- **Admin Use cases**
  - Add Song
  - Edit Song
  - Remove Song

- **Common Use Cases**
  - User, Billboard Database
  - Login

UML Use Case Diagram

Users can "View Top 100 Songs" to keep up with the most recent number-one hits and "Search for Songs" to easily find particular songs. "View Song Details" and "View Artist Information" provide in-depth information about songs and their authors when they "Listen to Songs". The "View Chart History" displays a song's performance over time. By adding, updating, or removing songs, administrators can "Administer Billboard Data" and "Update Chart Data" to maintain the chart's real-time accuracy.

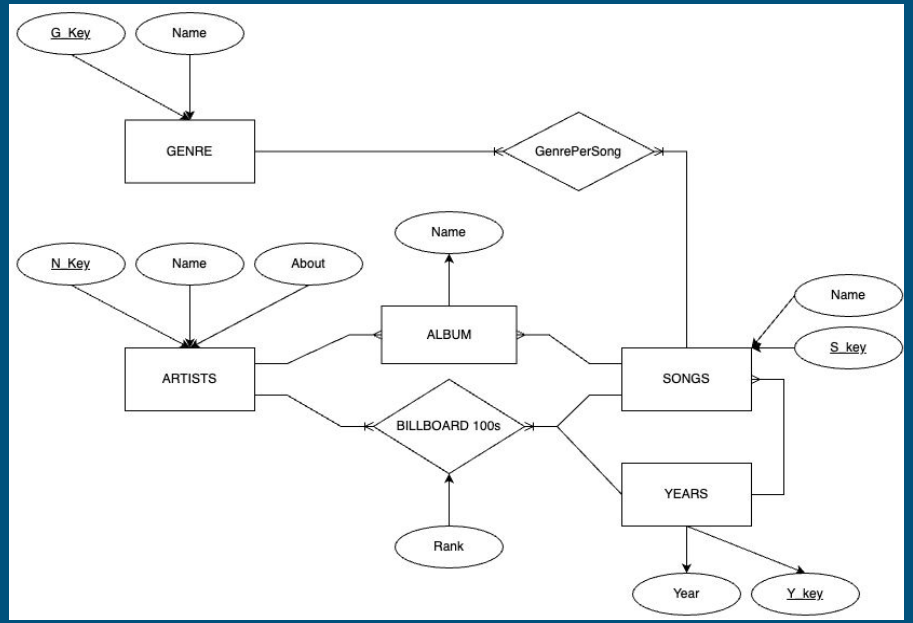


E/R diagrams

We have 6 entities in total

We have 2 different ways of joining artists and songs

Billboard Entity consists of 3 foreign keys



Relation specification

The GenrePerSong table is meant to help combine the Genre and Song table as these tables have a many to many relationship