



# **YEAR-END CHARTS**

## **Billboard Top 100 Songs**

---

By: Pranav Chakilam & Mark Soria



# Implementation Purpose

---

## **Provides insight into the billboard top 100 from 2006-2022**

- Collect data of billboard top 100 from 2006 - 2022, artists associated , albums associated, year they were created, and genres associated with them
- Use the collected data to edit, sort, and find specific songs, artist, albums, etc by accessing the billboard top 100 songs from 2006 - 2022

# System Description

---

## **Integrated Development Environment (IDE):**

- Visual Studio Code (VS Code)

## **Database:**

- SQLite3

## **Language:**

- FrontEnd: HTML, JavaScript
- BackEnd: JavaScript

## **Framework:**

- React/Flask

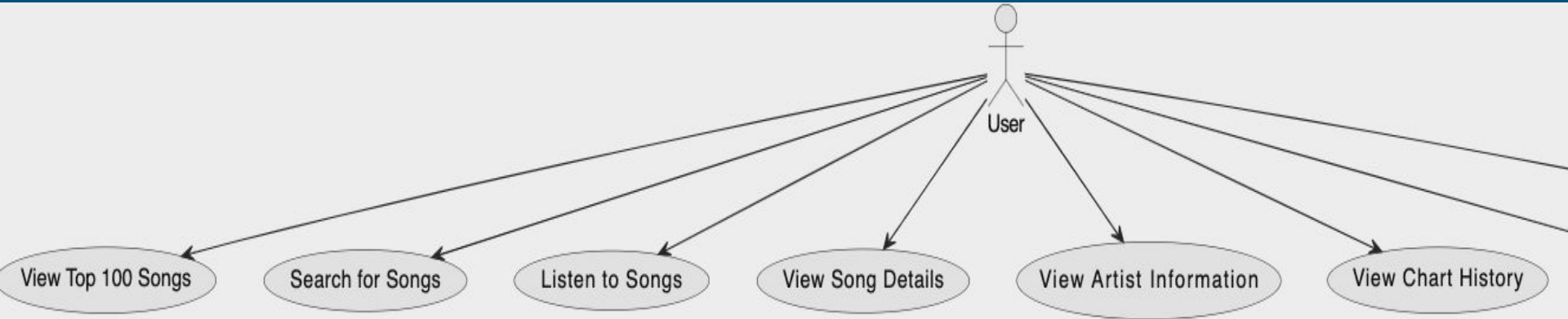
# Use-Cases

---

- **User Use cases**
  - View Top 100 Songs
  - Search for Songs
  - Listen to Songs
  - View Song Details
  - View Artist Information
  - View Chart History
- **Relationships**
  - Songs to Artists
  - Songs to Albums
  - Songs to Genre
  - Artists to Albums
  - Genre to Albums
- **Common Use Cases**
  - User, Billboard Database

# 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 and the year they were created in. "View Song Details" and "View Artist Information" provide in-depth information about songs and their authors when they "Listen to Songs".



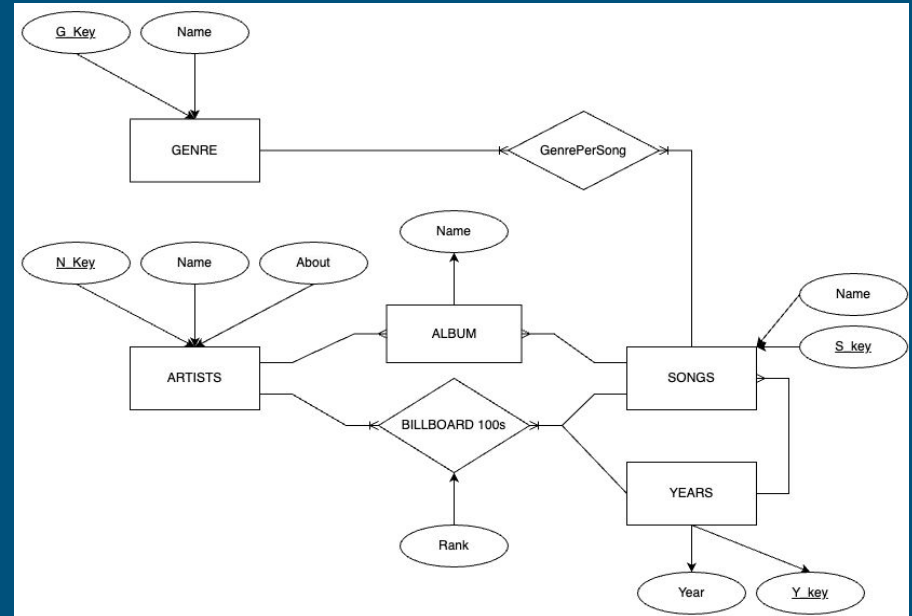
# E/R diagrams

---

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

