

# SongScribe

A black and white photograph of a young man with dark hair and a beard. He is wearing over-ear headphones and a white t-shirt. He is looking down at a black smartphone he is holding in his hands. The background is dark and out of focus.

Chris Delisle, Lilia Garcia,  
Matthew Long, Trey Verhelle

# About

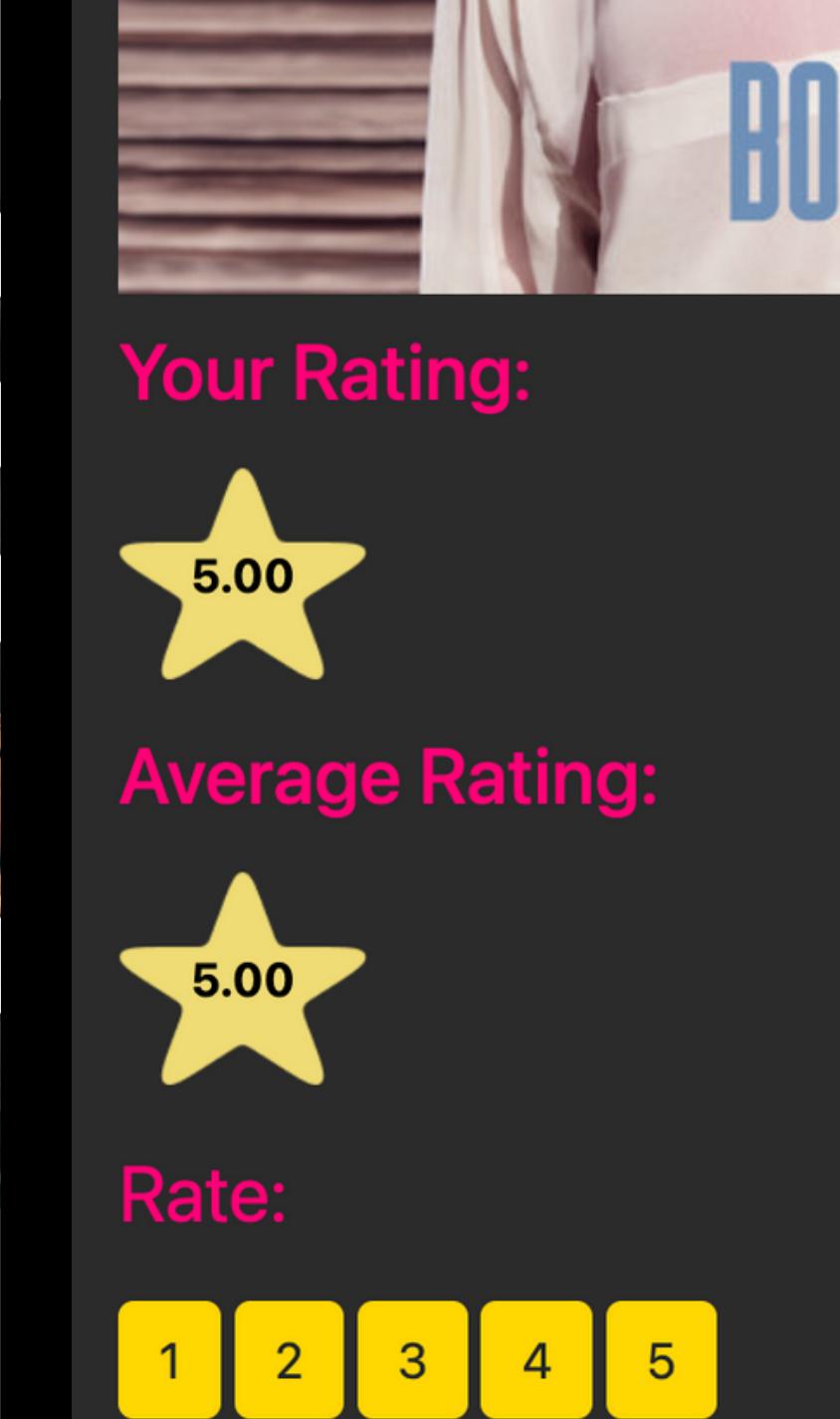
Song Slice is an app designed for anyone who listens to music. Users can rate, review, and discuss their favorite albums. This app aims not only to help you find your favorite artists, songs, and albums but also to help you connect with a community of others who enjoy the same music.

# Key Features



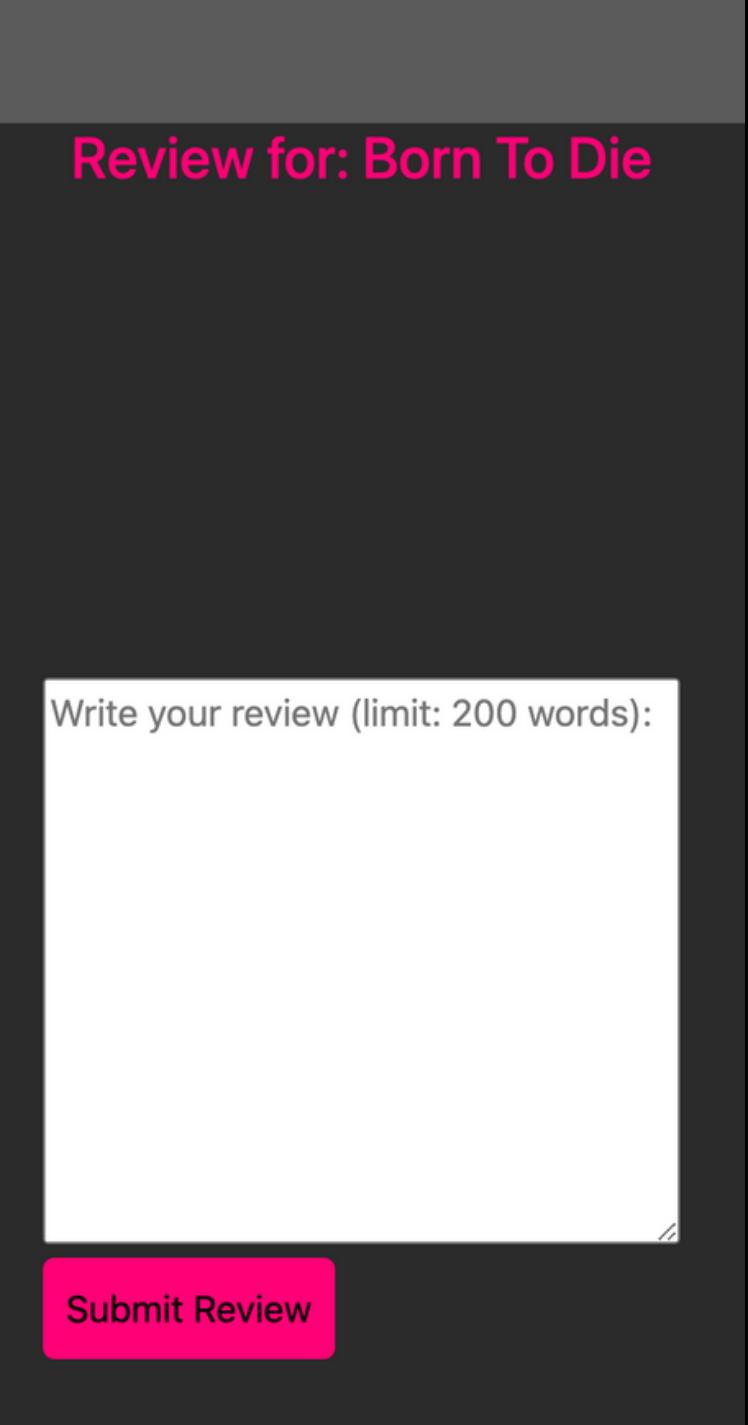
## Album Discovery

Upon logging in, users can explore a wide variety of albums on the homepage.



## Ratings

Users can rate albums 1-5, and see the album's average rating. Upon rating, it updates the average.



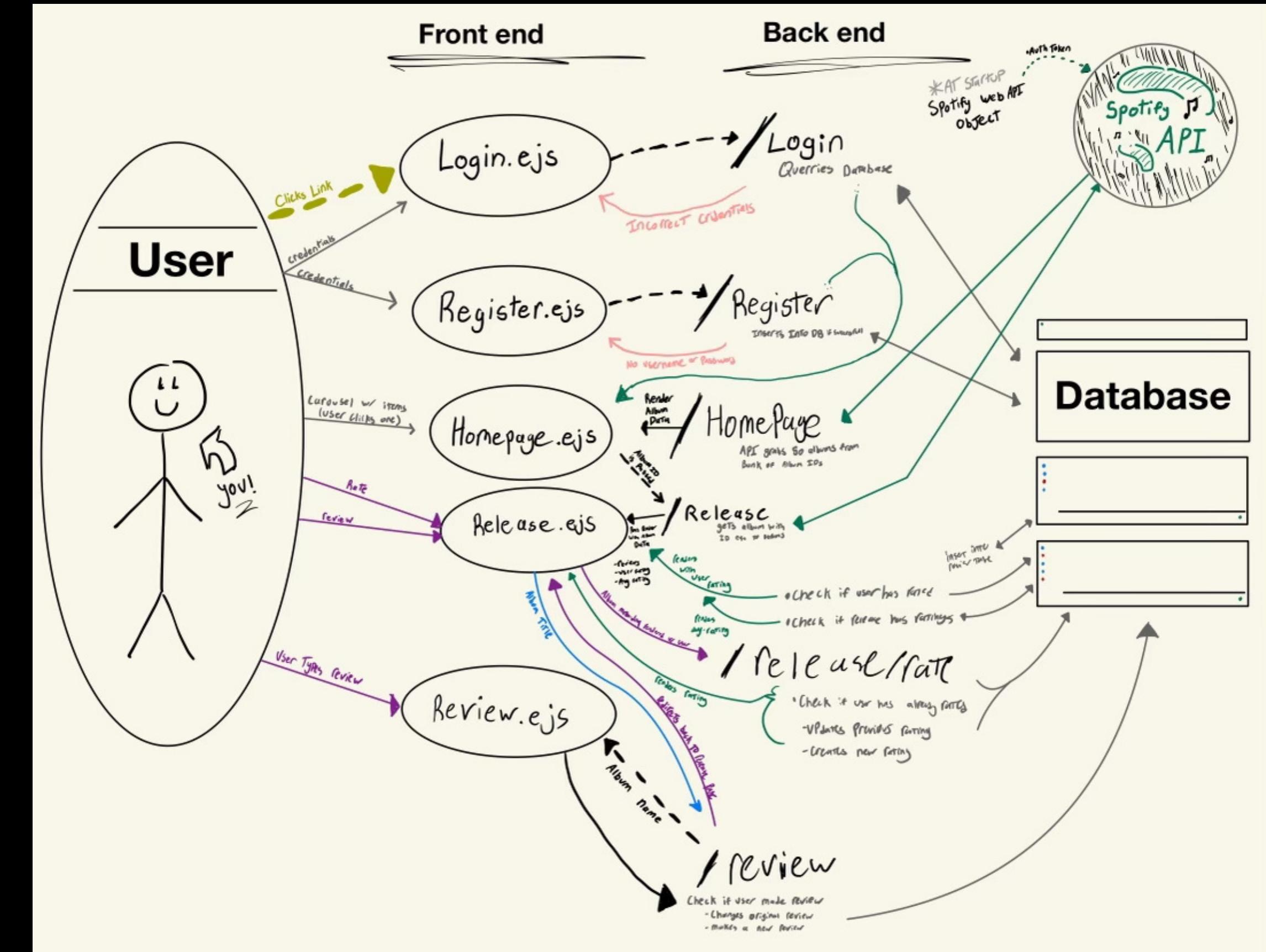
## Reviews

Users can add a review and leave their thoughts on the album. Under the album, you can view reviews left by other users.

# Tools Used

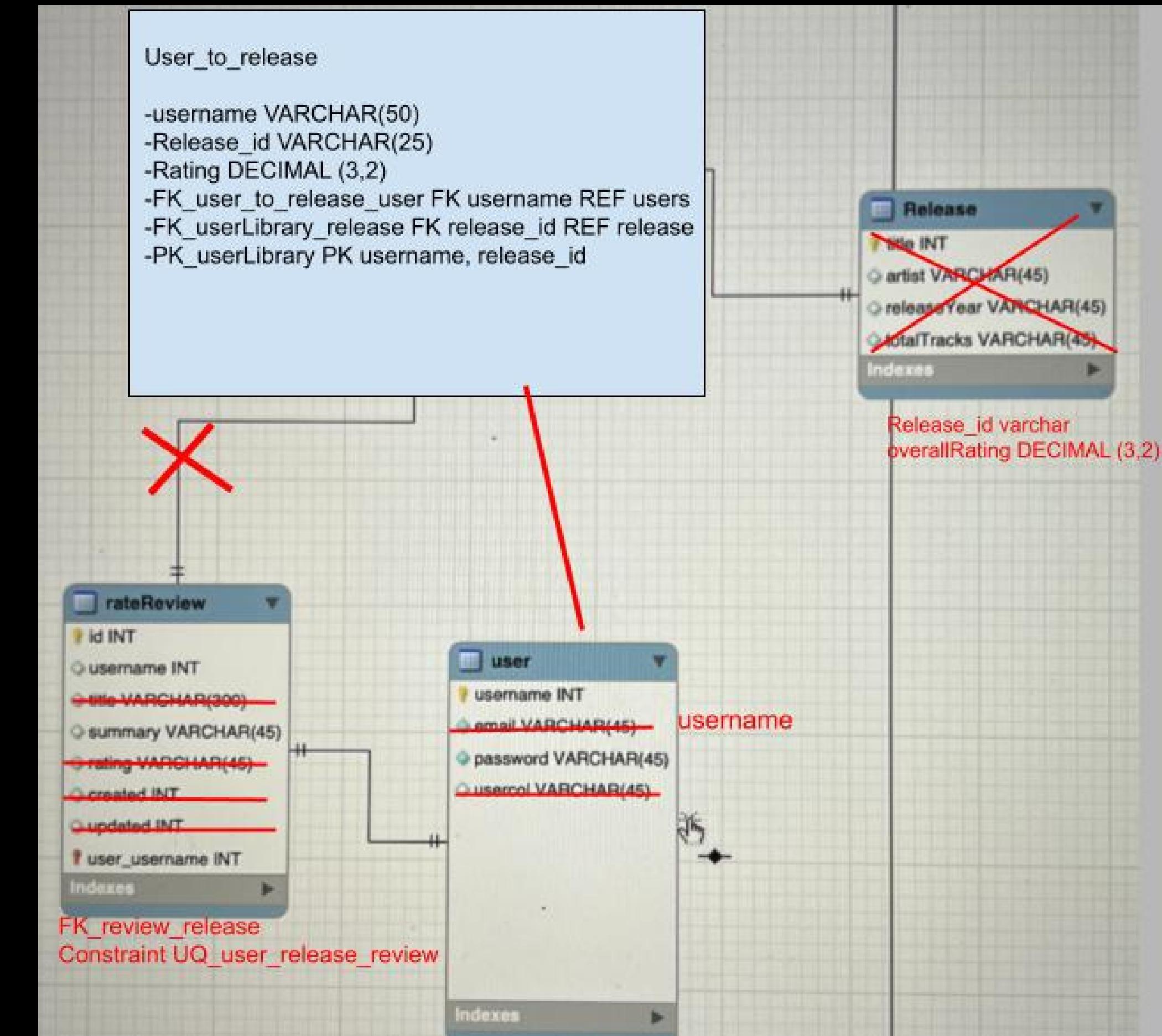
<b>Agile</b>	Hard to stick to. We weren't consistent because all of us have different schedules	2/5
<b>GitHub</b>	Gave us platform to merge, separate, and save code as we went along. Some issues with merging.	4/5
<b>PostgreSQL</b>	No issues, very easy to set up and use.	5/5
<b>Zoom/iMessage</b>	Flood of messages and hard to keep track, but served for quick updates. Zoom helped with urgent meetings.	4/5
<b>Figma</b>	Copy & paste hex colors, helped with CSS. Gave us a very good and accurate visual of how we wanted Song Slice to look.	5/5
<b>Spotify API</b>	Difficulty understanding the endpoint functions <a href="https://api.spotify.com/v1/albums/%id%">https://api.spotify.com/v1/albums/%id%</a>	3/5
<b>NodeJS</b>	Handled our server well, minimal tinkering required.	5/5
<b><u>Spotify-Web-Api-Node wrapper</u></b>	Made everything about integrating Spotify endpoints and authorization tokens easy	5/5
<b>Docker</b>	Overall helpful, but would have random issues out of nowhere	4/5

# Architecture Diagram



# Database Relational Model

(Includes changes made to our original model)



# Challenges



## Goals, expectations, and workflow

Too many features



## Testing

Adding features last minute caused complications



## Relearning content learned in class

Git management, docker, understanding promises and asynchronous functions



## Spotify API

Authentication tokens worked differently than lab9. spotify-web-api-node



## Scribework

Constantly flooded with messages at random times, so it was difficult to keep track of changes and write them down.

# Live Demo

SongSlice