Firebase Realtime Database

NoSQL server

Organised as JSON objects

Access to the database from client-side code

The data is stored locally on devices, so instead of making a request to the server, the server updates the data on all devices as soon as it changes on the database

Basically, there is no server, just organised JSON objects across the service

You can still decide what data people need access to at any given time – think passwords etc.

Horizontally scalable (MySQL is vertically scalable)

Good for large datasets and very fast

Structuring Data

JSON objects

No tables or records

When data is added to a JSON tree, it becomes a node in the existing structure with an associated key

In practice it is best to structure your data as flat as possible, to avoid large object pulls and to maintain security

For example, maintain an ID of a particular object and split the details across a number of JSON objects, to be able to iterate over certain data without having to download massive amount of data

Be aware that this can also lead to some necessary redundancy in the database, which will allow you to more efficiently find certain data

How will our data be structured

//this will be done in the authentication service I think

Accounts

Username

ID

Email

Password

Account Data

ID

Username

Name

Bio

Time on platform

Top Artists ID Array

Friends

List of other account IDs

Similarity Score etc.

Artists

ID

Links

List of Artist IDs they are related to

Genres

List of Genres they are included in

Songs

List of Song IDs they have credits on

Info

Name

Birth Year

Age

Death Year

Key Moments

Moment n

Year

Description

Rating Score

Songs

ID

Title

Release year

Bands

ID

Year Started

Year Ended

Members

List of Artist IDs

Tools

There are tools that can be used such as ReactFire to make some operations easier to interact with Firebase, but it seems like our application will begin simple enough that these will not be necessary

Scraping data from the web

Puppeteer with node.js?

Could just program it seen as we just need the certain sections of a Wikipedia page

Later we can brush up the more important ones

Spotify API can provide most of the other data we need and that can be saved to the database very easily

Firebase Cloudstore

It does have simpler code associated with it

The JSON objects are stored in Documents, which are stored in Collections

In fact it’s more like there is no JSON, but documents look and act a lot like JSON objects

Again, the idea here is to keep the tree as flat as possible, with large collections of small documents

However, it requires this data flattening much less than the realtime database, as you can create more direct references, similar to a normal database, instead of having to download tons of data

Collections can only store documents, but documents can hold anything, including other nested collections

Data should be structured with a lot of subcollections, as it allows for easy and powerful queries, but the limit here is that subcollections are difficult to delete

Data that may be deleted should be kept in documents

You can have transactions and batched writes, which allow you to read and write from and to up to 500 documents at once

In both firestore and realtime db, there is a lot of js and anonymous functions, so it should be able to integrate with the rest of the backend

Final Design

Accounts

-> username Docs

-> ID + email

Profiles

-> Id docs

-> user info

Artists

-> ID docs

-> Info Doc

-> info on artist

Bio scraped from Wikipedia

Name, DOB, etc.

-> Key Moments Doc

-> key moment n

-> Songs Doc

-> song IDs

-> rating score

Songs

-> ID Docs

-> song info

Release year, title, genre, etc.

Bands

-> ID Docs

-> Member Docs

-> artist IDs

-> band info

Name, years of career

-> Key moments Doc

-> key moment n