**CS3200 Project Final Report**

**Project topic:**

**Music Database System**

**Group: ShenYZhaoZ**

**Group Member : Yuxi Shen, Zhiheng Zhao**

**Professor: Professor Kathleen Durant**

**Abstract Description**: Users can experience fully personalized music services and recommendations on this music software the whole time by this database.

1. **README section**

**Description of the Application:**

We plan to create a page to enable users (members and administrators) to interact with the database for searching songs and albums from global musicians by Python. There are two types of Users : members and administrators, both of them have unique user\_id and user\_name. Administrators have access to registered users’ personal information and search history. First, users can CREATE their account with a unique username. After logging into the account, users can either LISTEN to the recommended playlists or SEARCH for songs/albums/artist to listen. Then, the search panel could READ users’ search information, and DISPLAY the result. After that, users could LISTEN to the song or CREATE a new playlist folder with it. A user can UPDATE their personal data. Admin can UPDATE the database information or any other user information. Admin can DELETE a user or song related information. Also, the database can recommend songs to users based on their search history.

1. **TECHNICAL SPECIFICATION**

* Database Language
* --- MySQL
* --- Software: MySQL WorkBench
* --- Application Language: Python
* Justification for SQL Database:
* ● MySQL Database can support complex queries to be performed on the data stored in the database.

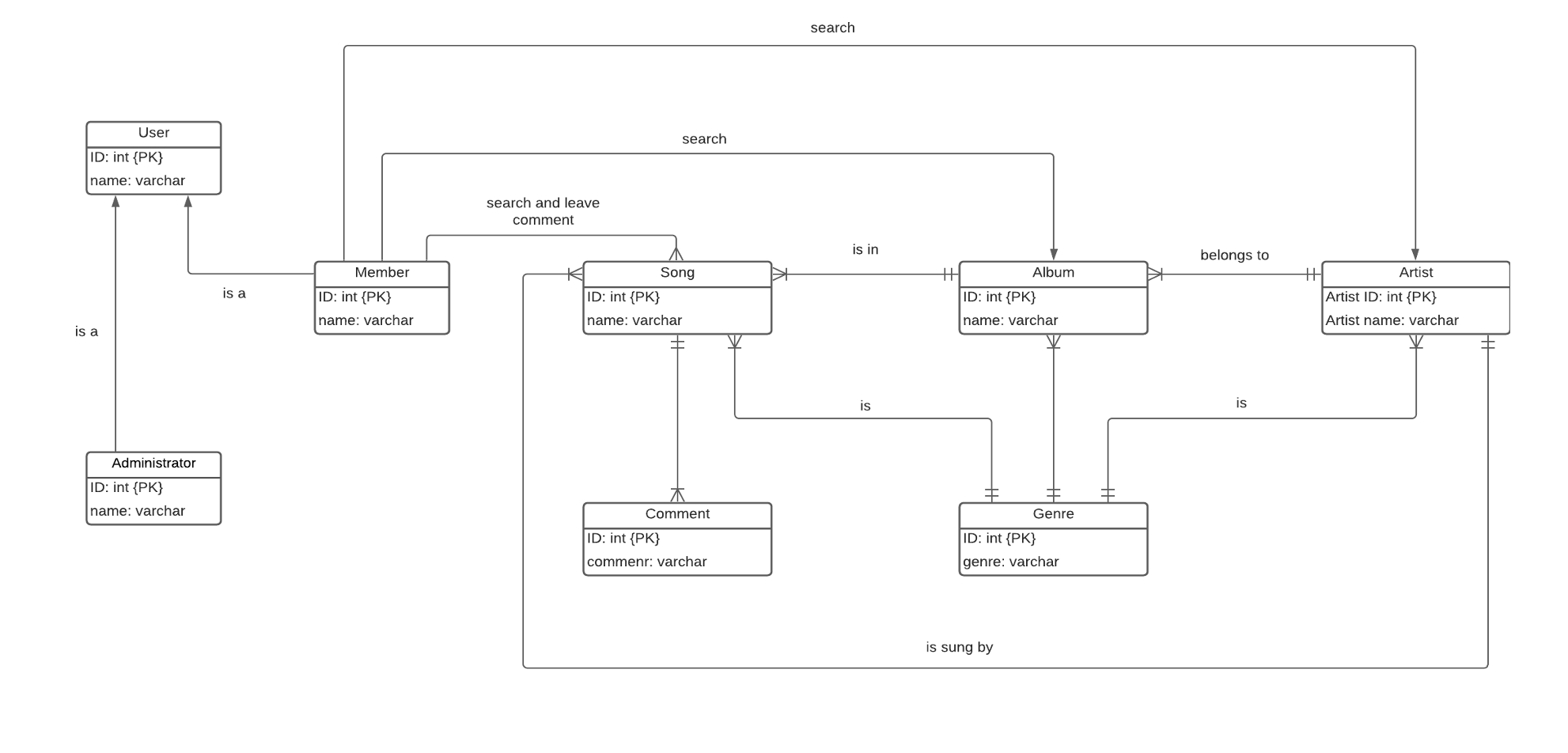
* ● It is important that the database responds quickly to the user requirements.

* ● MySQL also supports Replication, so the database can be hosted on multiple servers which ensures workload balance and scalability as the users of the applications increase.

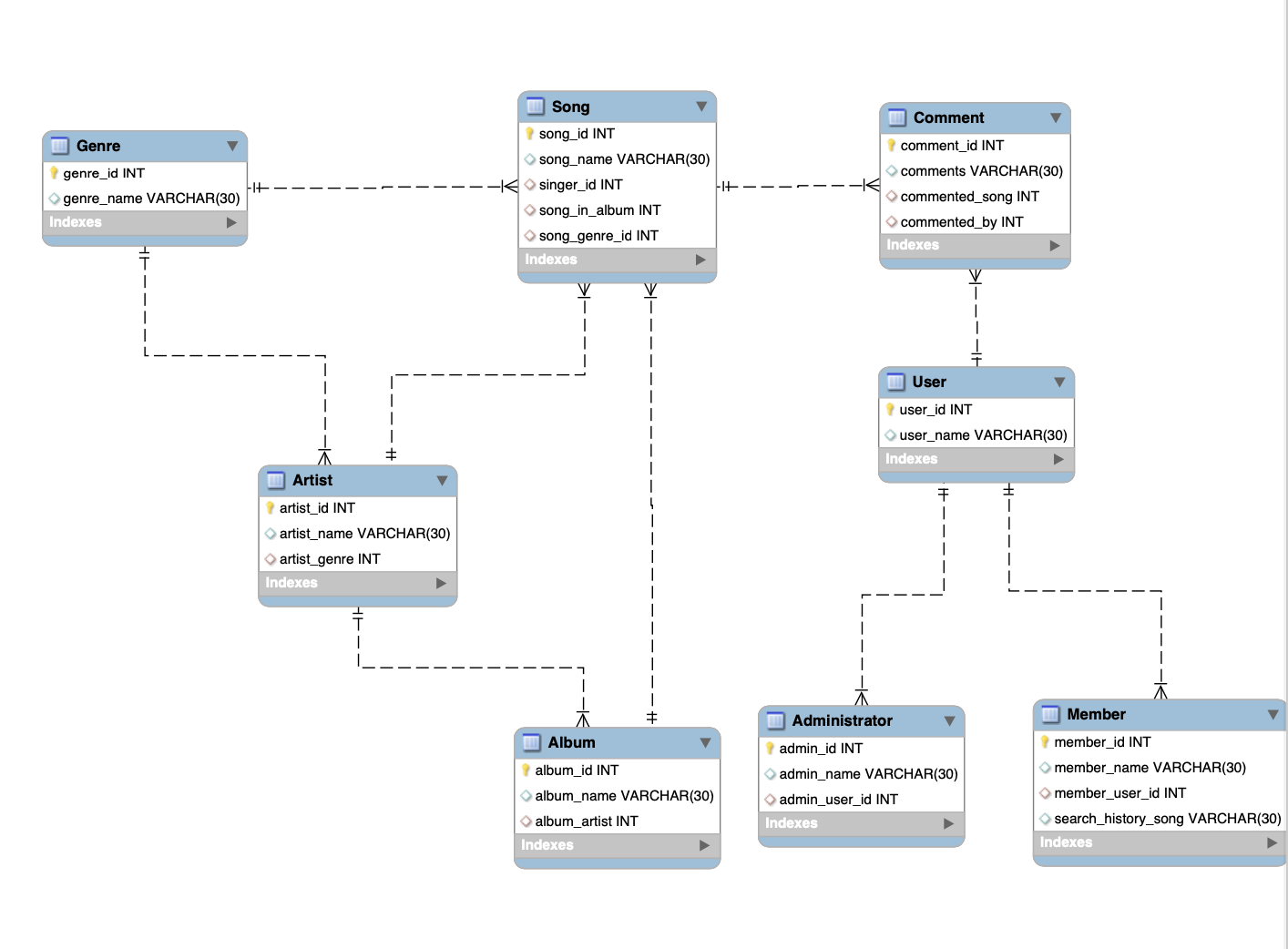
* ● NoSQL databases are slow in providing response if the queries are complex.

* ● Also, NoSQL databases are suitable if schema of the database are dynamic. In our application schema is not dynamic.

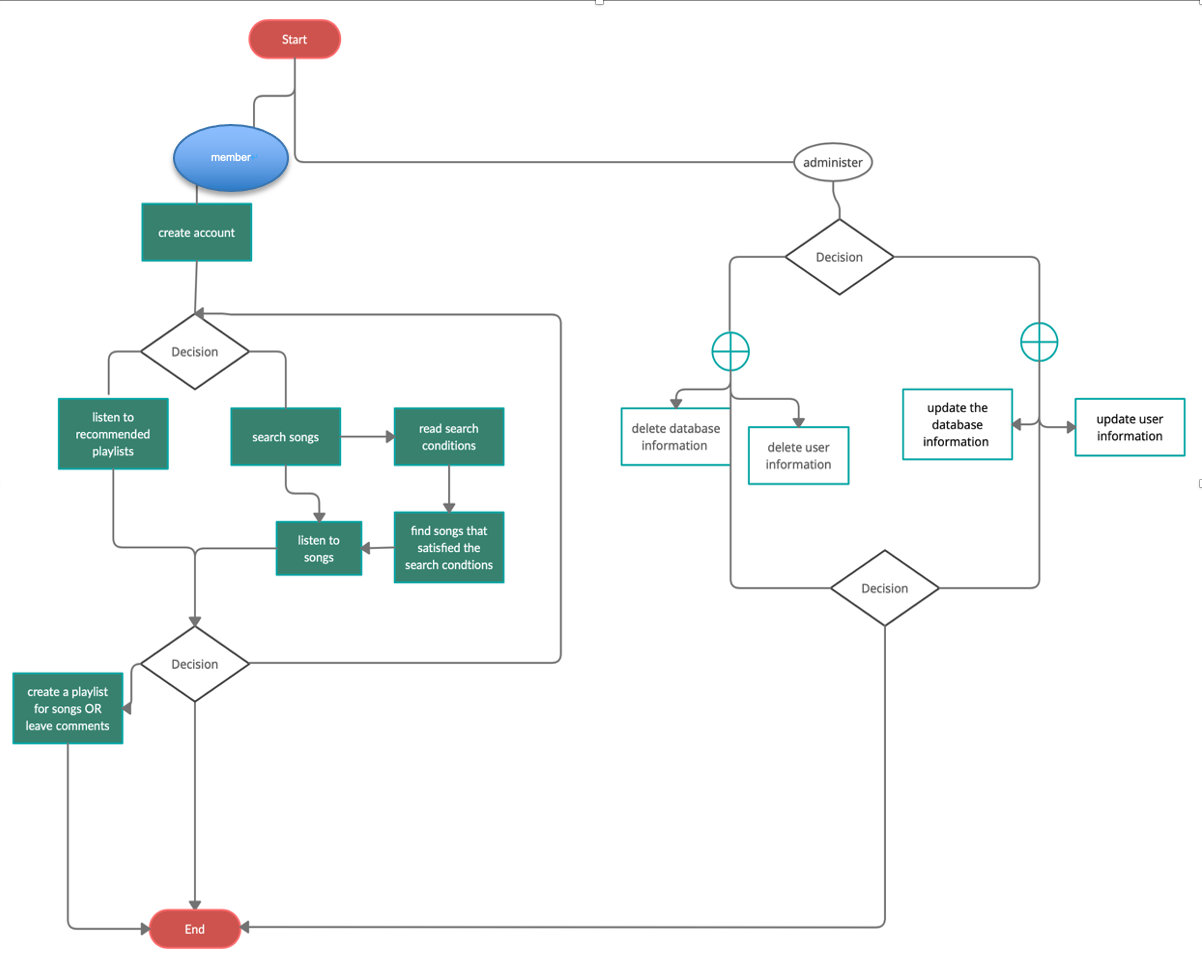
1. **CONCEPTUAL DESIGN**



1. **LOGICAL DESIGN**



1. **FINAL USER FLOW**



**User interaction**

-- What users can do?

If he/she is a member

1. Create account

2. listen to recommended songs

3.Search for songs/albums/artist

4.Store their searched song

5. Leaving comments under songs

If he/she is an administrator

1. delate database data
2. delete members’ information
3. update database data
4. update members’ information

1. **LESSONS LEARNED**

In the process of completing this project, we not only learned a lot of professional experience related to technology but also realized what kind of work and life we might experience if we truly regard data scientists as our career development goal in the near future.

To create the database, we utilized MySQL to create tables and necessary commands. The project offers us a perfect opportunity to create commands by ourselves, but rather than writing required commands with the requirements in assignments. It is a difficult but also interesting mission for us in the beginning. For example, it is simple to write a trigger in homework7 by guidance, but it is hard to accomplish when we have no clue to track, i.e. there is no specific requirement. Although we have encountered some setbacks, we are also more proficient in writing SQL commands and more efficient in creating UML graphs to promote database creation.

Besides technical expertise gained, we realize that time management and building plan are also important skills. We assign tasks according to each other's best skills, which greatly improves our work efficiency. At the same time, we are also more familiar with the usage of GitHub and google doc. As we all know, Github provides the function of sharing and saving changes online, so we use GitHub as a communication tool and a backup at the same time.

Our goal is to create a personalized music database similar to Spotify, but in the process of achieving the goal, we found that achieving each word recommendation will require a larger user database and user usage records. Therefore, we can only establish a recommendation mechanism that is not very complete. For example, based on user search records, we recommend songs released by the same singer or songs with similar genres. The database we build can at least allow users to find and collect a series of basic functions.

Although our group has invested a lot of time and energy to implement our fantasy music database, we are not very confident in the command to delete database-related information that only the administrator can authorize.

Overall，Thanks to this project for giving us a free environment to create the commands and functions we need. This is also inseparable from the teaching assistants and the instructions of the official SQL website.

1. **FUTURE WORK**

For the database we designed, we plan to use it to perform simple searches, update and add data. In addition to the technical plan, we also hope that through the design of this database, we can figure out what functions a fully established music database software should have. Because just as a user, we don't want to think so deeply as a developer. And only like this, we have to be a developer for some reasons, will we realize what skills and technical support are needed to build a database.

We hope that even after the end of this lesson, we can continue to add instructions and tables to the project to make this database more complete, such as providing users with the service of freely editing song names and nicknames.