**PROJECT FINAL REPORT:**

**MusicDB**

*Group 7:*

* *Duy Vu*
* *Nam Luu*
* *Lee Jaejin*

1. **Provide a README for running the project. Specify all libraries, software, etc. needed to run the application:**
   * **The project is also being hosted at:** [*http://cs3200project-myneu.rhcloud.com/public/index.html*](http://cs3200project-myneu.rhcloud.com/public/index.html)
   * All the files that is needed to run the project are in the front end code assignment zip folder. (I submitted both the front end and the back end code in that zip)
   * The user computer need to have NodeJS install
   * Then unzip the frontend assignment code zip, go to the directory and run the server,js file
   * This file can be run by opening a command prompt window at the directory and run ‘

🡪 node server.js. Then go to the browser and type 127.0.0.1:3000

**Please let me know if you have any difficulty in running the application**

1. **Provide a final ERD diagram of your project:**

**Song:**

* \_id : String
* title : String
* artist : String
* album : String
* year : String

**Playlist:**

- \_id : String

- userId : String

- songIds : [String]

**Comment:**

- \_id : String

- songId : String

- username : String

- content : String

**User:**

- \_id : String

- firstName : String

- lastName : String

- username: String

- password: String

- Email : String

- Roles : String

- Song : [String]

1. **Provide the final user flow of the system. List the commands used to interact with the system:**
   * Go to the website: 127.0.0.1:3000 or [*http://cs3200project-myneu.rhcloud.com/public/index.html*](http://cs3200project-myneu.rhcloud.com/public/index.html) *(already has admin/admin as admin and alice/alice as regular user)*
   * Register or Login
   * Register with an admin username will register you as and ADMIN. Any other name is regular user.
   * Register/Login will lead you to the profile page where you can update user info
   * Home tab is used to search for song on Spotify. Then add/delete the song to library by clicking on the song’s image
   * Library tab display the user songs. User can delete or select song
     + Select song will display the song on the bottom bar: user can then click on the 3 bar symbol on the bottom bar to see/add/delete comments on the song
     + User can add song to playlist (if there is a playlist) with the ‘+’ sign
   * The playlist tab on the Library page is for creating and deleting playlists
     + Clicking a playlist will show you the content of the playlist
     + User can delete/reorder the song in the playlist content page
   * For ADMIN users:

Admin tab is available: an Admin can create/update/delete **any** user. Create/update/delete **any** song. Create/update/delete any comments

1. **Provide a lessons learned section:**

* We changed from using mySQL to mongoDB.
* We had haft of the project done in mySQL. As you can see in the **databaseInfo** folder and the **public/server/model/…mock.json** files. We started with the mock files data, then built the MySQL tables. We has created many procedures and get the project to work in MySQL at one point. Until we decided to add more functions such as (playlist, duplicated songs in playlist, multiple comments for songs)
* Then we decided to switch to mongoDB. The reason is as we progress and adding functionalities for our project, we changed our schema many times (adding comments schema, duplicated songs in the playlist...).
* Every time we changed the schema in mySQL, we have to split up the tables and it increase the **complexity in designing procedures** to modify the data structures in multiple table in the database. Not to mention the **time constrain** of rewriting the whole front end in many cases.
* Since we didn't know if there will be any more changes to the schema, we decided to use mongoDB. Because the mock data and MongoDB are all in the same JSON format, it is much easier to modify the structure of the data to match the front end.

1. **Future work**
2. **Planned uses of the database:**

- We planning on using the database for managing our own music library across the music platform. In order to do that, however, there are functionalities that needed to be implement

2. **Potential areas for added functionality**

- Expanding the number of services: it is only Spotify for now, but we can easily adding searching for songs on others services

- Adding importing song’s info via excel/csv files, this is the best way to push the user library to the cloud.

- Along with that automatic import process is the auto matching to match the imported song with the service database.