Rationalizing UML Diagrams

Class Diagram:

The ‘User’ class consists of the login and signup information of the user as the attributes. A user must already be signed up and will have the ability to rate, search, sort and view their homepage after logging in. The ‘MusicDB’ is the database for all the music, where we will pull information about the songs from, here is where we get the music API and a song’s database from, and where the database is updated to reflect new music. The ‘Search’ class has the attribute of the search keywords and it is reflected in the search page, search list and goes to the song database when a specific song is clicked. ‘Search’ inherits from both the user, by taking the user's search words, and from the music database, which allows for a search page and list to show all the matching songs. There is also aggregation hierarchy represented in the ‘Rate’ and ‘Sort’ classes. Without a user, the search and rate classes would not exist and we can delete these classes but cannot delete the user class. Both classes get the song’s database, allow the user to rate or sort the song, update the rating or folder, and then update the user’s homepage.

Data Flow Diagram:

The data flow diagrams represent both the rating of music and sorting of music. These are more straightforward and easy to understand. We do not include the login/signup aspect for both these diagrams, since a user on our application must already be logged in to rate and sort music, including it would be redundant. A user will search a song's name, and that will be sent to the music database to display the song matches, which will be sent back to the user. Then, the user can click on the title of the song, and all of the song's information will be displayed from the song database. From there, a user can rate or sort the song and it will then update the user’s own database and homepage to reflect the rating or sort.

Sequence Diagram:

The user can search for music and the website will return the results. Then, a user is able to view the songs and click on them, then the website will return the song’s information from the database. The user can rate or sort the song into a folder. The website will be updated anytime there is a change in the user’s database to reflect that change, like when a rate or sort occurs. The viewing of songs, clicking on song titles, rating and sorting of songs are all optional features and up to the user to decide if they want to do them. The searching of songs is not optional since that is the bare minimum our website will provide users, even if they do not rate or sort music they will still likely search. And lastly, the updating of a user’s page is also not optional, anytime a change occurs, the user’s homepage will be updated. All of these sequences can be included in an overall loop depending on how many times a user uses these features.