**Classes**

**Server**

* **Spotify monitor**

Spotify Monitor will be constantly listening to the Spotify API. This ensures that whenever the client is listening to music, we are able to store that data onto our server.

* **InProcessor**

InProcessor will process the incoming data from the Spotify API. The data from the Spotify API needs to be processed before storing, so the InProcessor filters out the unnecessary data and stores the rest in a JSON file.

* **OutProcessor**

OutProcessor processes the data which has been requested by the client and is currently saved on the server. The data needs to be analysed by the the OutProcessor so that some meaningful data can be sent over to the client.

* **Store?? (Database Handler)**

The DatabaseHandler will process requests made for access to the database. This includes sent information from the SpotifyMonitor needing to be saved and data requests made by the ClientHandler.

* **ClientHandler**

The ServerHandler will process requests server side sent by the client. It will handle the request and start the process of pulling data and analysing it. It will also handle sending the information back over to the client once it has been analysed.

**Client**

* **GUI**

GUI will be responsible for displaying the data sent from the server side. It will provide the interface for the user to interact with.

* **Server Handler**

The ClientHandler will send over requests made by the client over to the server. ClientHandler will also be responsible for receiving data and handling it sent from the server.

**Methods**

**Sportify Monitor:**

* **Monitor**

Monitor is a method which will be running constantly. Its job is to listen out to any activity recorded by the Spotify API

* **SaveData**

SaveData is a method for sending data from the Spotify API to the database on the server.

**OutProcessor:**

* **AnalyseData**

AnalyseData is a method for analysing the data on the server in order to make it more meaningful. Depending on the information we are going to actually show to the client, this will more than likely end up being multiple functions. (Fit for different types of analysis)

* **GetData**

GetData is going to be used for sending requests to receive data from the database .

**InProcessor:**

* **GetData**

GetData for the InProcessor will be used for getting data from the Spotify API

* **ProcessRawData**

The data received from the Spotify API will be raw and will need to be processed before saving it to the database. This method will be responsible for converting the relevant information into a JSON file and saving it to the database.

**Store?? (Database Handler):**

* **GetData**

GetData for the DatabaseHandler will be used for getting data from the database.

* **SetData**

This will be the method used in order to save data to the database.

**Client Handler:**

* **ReceiveClients**

Receive clients is the method responsible for listening out for clients and for accepting and consequently threading them.

* **SendData**

SendData will be used for sending the processed data back over to the client once it has been requested.

* **HandleRequest**

HandleRequest will be used for handling the requests made by any client. This will start the process of getting the relevant data from the store in order for it to be analaysed.

* **StoreData**

StoreData’s purpose is to save clients’ data such as account information.

**GUI:**

* **FormatData**

FormatData is a method that will be used for formatting the data sent from the server. It will process the data into relevant categories ready for it to be displayed.

* **DisplayData**

This will be used for displaying the data in a way such that it is easy to read and understand. Also, it will be responsible for displaying the data in some aesthetically pleasing way.

**Server Handler:**

* **RequestData**

RequestData will be used to send requests to the server.

* **SendData**

SendData will be used for sending data from the client to the server, such as personal information or account details.

* **RecieveData**

ReceiveData will be used for processing the data and requests made by the server.

**These are the diagrams we need to complete (Claim on Github so we don’t do any twice)**

**UML**

* Class Diagram
* Sequence Diagram
* State Machine
* Use Case Diagrams