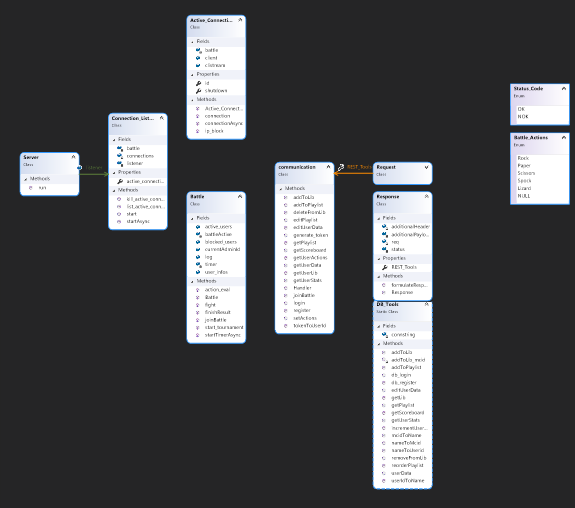
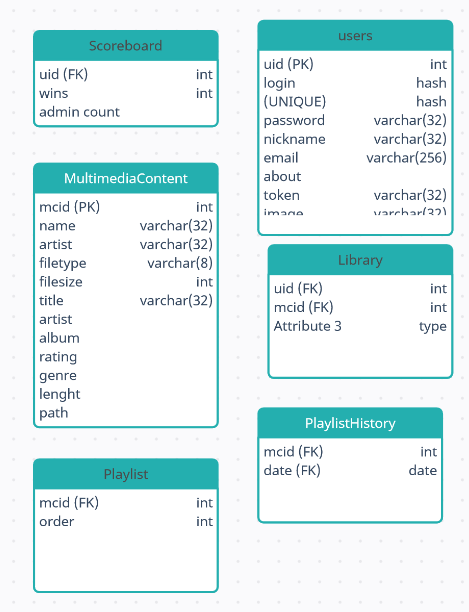
Technical steps

The first part of the development was creation of class diagrams and database diagrams. I used the built in VS class designer and free online tools for this purpose.



The coding part of the development was divided into 3 parts: Database Access, Battle Logic, and Connection.

Database:

I used the recommended Postgre portable database. The connection is made using npgsql library. I decided to use non-async static methods when working with specific database functions. While focus on static methods enabled a quick workflow the unit tests were significantly impaired due to the fact that static methods cannot be overwritten by mocking. So most test were done during the programming and relied more on curl scripts as well as test operating on the actual database.

Battle Logic:

I’ve put an instance of the class maintaining all the processes regarding to battle in the class maintaining connections. The pointer of this class is shared with every other method which interacts with the class. This enables a simple way to make sure that all users partaking in a battle, do so in the same instance. Testing of this class proved to be simpler, as the reduced reusability of the methods didn’t call for the static workflow.

Connection:

The majority of this part was already done in the previous assignment. I’ve rewritten the code to include slightly less classes than before. Just like before listener creates active connections which run async – this time, it was absolutely necessary to do so, due to the fact I wanted the user to wait for the response after the battle which – with more users – would make battles impossible if the connection wasn’t async.

The requests and response classes are responsible for transforming the input into the required format (accordingly to their names). The method “handler” of the “Connection” class takes adventage of that fact and uses variables from the request instances for the decision making and parses to the respective function.