Class diagram explanation : server part.

We chose to use spring boot server, which requires several classes to run the server and the database. ServerApplication class is a main class for the server part, which launches server itself. It does not, however, control any game logic, or database logic, but it was included because without it simply nothing would be running. We also included UserApi class and (not yet released) SkinApi class. UserApi includes in itself all the possible paths we need for must haves and methods for them. For example, it is fully responsible for authentication, getting maximum score, and getting information on a specific user. To get all of them, we have made special request classes, which contain information on a specific request to make (username + password to create a profile for example). We also have similar structure for returning classes, which is UserResponse, which has to implement interface due to how Spring repositories work. Similar story with SkinsApi class and all the related classes.

Next choice of classes to add was on database classes, since they are also an important aspect of the game. Both databases extend spring JpaRepository class, with custom methods added (and custom prepared statements). Documentation link to JpaRepository is added on the file. Nothing much to say, except that JpaRepository requires you to have interfaces for custom queries, so that had to be included as well.