# Design Description

The first thing to do during the design is to connect the project to the API given. In this case, I used Retrofit2 as the library to process HTTP requests and response. After analysing it, I found that there are a huge number of cards (around 10,000) in the database even though filtered the ones which don't meet the rarity requirement. The API can only return 100 cards in one page so multiple requests have to be made and consolidate.

With data fetched the next thing to do is to pass the data from the loading activity to the gameplay activity. Using Intent and making the data Object Parcelable cannot work here since the data I would like to transmit is too heavy and over its 1M limit. Therefore, I create a class extends Application as a global variable to store all the cards data.

In the game activity, I used Glide to acquire the pictures over the internet and load them into the ImageView. Another problem found here is that some links the API provides are invalid, so I have to filter the unreachable picture and reload another one.

In terms of game logic itself, it's not complex at all. To choose two distinct cards with different rarity every time, I classified all data by rarity when fetching them. Then the program randomly selects two different rarities and then randomly select a card in each rarity as well.