The connection starts with a ping message sent by the client to the server, which answers with a pong message to establish the linking. A timer between the interaction is set and if it elapses an error message is shown. The interaction between client and server can be divided in three major diagrams:

- The login phase: after initializing connection the client sends username, number of players and game mode, which compose the same json and the server checks the correctness of the parameters and answers with an error or an ack. Finally the client inserts the mage corresponding to the deck of cards he wants, which is sent to the server and checked. At this point the client is logged into a lobby which waits until it's full and when this happens, the game is created and the server sends to the client all the information about the board
- Turn initialization phase: at the start of every turn each player chooses a card in a preconcerted order, so the client inserts the id to the corresponding card he wants to play and once sent to the server it checks whether the card is valid and responds with an error or an ack, waiting for all the players to make the choice. In the end the server sends the order for the upcoming turn.
- Player action phase: following the order received, each player can choose which cloud card to use to refill his entrance, then he can move three times a different student by inserting the number and the position, which are checked by the server, which responds with errors or acks and at the end of the movements it sends the updated board. Once received the board the client inserts the steps to move mother nature, which are checked by the server which responds with ack or error and, when correct, it shows the board updated after completing all the operations. If the game is pro the player can insert the id of a character card he wants to use which is also going to be checked by the server.

Every error message has the following json structure:

```
{
"name": {
  "message": " ",
  "code": XXX,
}
}
```