PEER REVIEW

Project Name:	Prova finale di Ingegneria del Software
Reviewed Project Members:	Simone Cervini, Ludovica Cova, Davide Fugazza
Review Author:	Aleksandro Aliaj, Leonardo Cesani, Matteo Colella
Review Date:	05/06/2022

Strong Points

Title	Annotations	Suggestions
BPMN diagram	Very clear representation using this language	None
Messages	The messages clearly describe the interaction	None
Simple protocol	The protocol is easy to understand	None

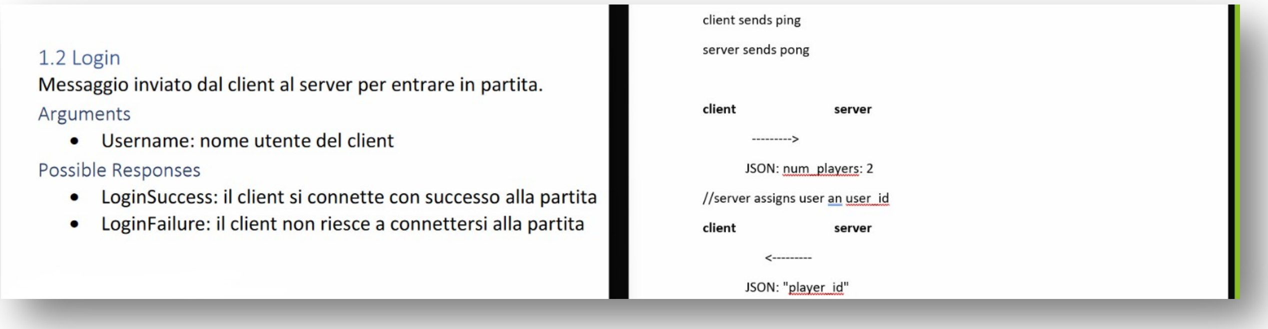
Weak Points

Title	Annotations	Suggestions
Sequence Diagram	The representation of the sequence diagrams seems to contrast with the documentation file	We suggest looking back at the diagram making it more coherent with the messages, for example, in the diagram we cannot see the username shown in the messages. We linked an image below [1].
Error messages	Lack of different error message types	We suggest adding more error messages to make debugging easier
Sequence diagram #2	The sequence diagram illustrated describes a specific game for two players	It would be better to describe a game in a more generalized way.
Error messages #2	It is not clear what happens after an error occurs	We suggest describing how the interaction proceeds after an error

Messages	Implicit messages codes	It would be better to explicit how the server and the client recognize the type of a message

Comparison

- The most significant difference we noticed between our diagram and the one reviewed is the login phase due to the fact we want to implement a multi-game structure, every player who logs in can choose username, number of players and game type and the server checks all this information together, answering with an error or an acknowledgement message inserting the player to a lobby. On the other hand, the diagram we reviewed seems to concede only to the first player to decide what kind of game he wants to play.
- We implemented many types of messages described by a particular code to make it easier to recognize them. The group reviewed choose to have less messages and not to explicit a code, this could result in an easier implementation but harder debugging.
- We also noticed that there’s a difference in controlling client status: we implemented a ping pong thread to check whether a client is online or not by exchanging continuously messages in parallel to the main client and server thread. By reading the group documentation we understood that every message sent by any of the entities is answered with a pong to acknowledge client or server status.



1.2 Login

Messaggio inviato dal client al server per entrare in partita.

Arguments

- Username: nome utente del client

Possible Responses

- LoginSuccess: il client si connette con successo alla partita
- LoginFailure: il client non riesce a connettersi alla partita

```
client sends ping

server sends pong


client              server

      --------->

   JSON: num_players: 2

//server assigns user an user_id

client              server

        <---------

   JSON: "player_id"
```

Image 1: Comparison between documentation and diagram