

What we fixed:

- **Error 1:** The password length was not restricted
- **Fix 1:** We limited the length on the input of the ID and password to 100 characters

- **Error 2:** The increase and decrease in the amount of each action was unlimited
- **Fix 2:** We limited this quantity now to 10 digits

- **Error 3:** When the client entered a file name which was on multiple lines, i.e. the contents of a JSON file, the client crashed.
- **Fix 3:** Now when the client wants to input a file it gets redirected to a filechooser menu where only JSON files can be submitted.

- **Error 4:** We allowed for negative delays, which were translated to a delay of zero seconds
- **Fix 4:** Now, on validation, the whole file gets rejected if the delay is set as a negative number

- **Error 5:** When an inputted port was a valid port, but not that of the server, the client crashed
- **Fix 5:** We catch the exception to an incorrect port, with a 'ConnectionRefusedError' and tell the client to check the port and try again.

- **Error 6:** When an action had the format of a number, followed by a letter, our program would continue the action causing an error
- **Fix 6:** We implemented a regular expression check on the input data to make sure that it's strictly correct

- **Error 7:** Inputting a file with the wrong format wouldn't load
- **Fix 7:** It's fixed with the fix of error 3, where the user can only choose JSON files as input from the filechooser

- **Error 8:** Increases and decreases with decimal values, crashes the client.
- **Fix 8:** Decimals are now allowed, and checked with a regular expression.

- **Error 9:** Uses the path from the place it was launched; which causes issues on folder structure.

- **Fix 9:** Fixed by the filechooser we added.
- **Error 10:** If the previously entered file had an extra field, it would be rejected but cause an error when the next file is entered.
- **Fix 10:** Now when an incorrect file is inputted the error message is displayed and client closed. The client must be re-run to send the next JSON file.
- **Error 11:** When a second client connects with the same ID but a different password, both clients crash and the server raises an error but does not crash.
- **Fix 11:** There actually were 2 errors occurring here. Firstly, there was an error occurring in the original client and the server. This was because the second client with wrong password connected to the server but did not increase the number of connections with the ID. However when it disconnected it decreased the active connections with that ID from 1 (the original client to 0), in our case this deletes that entry from the active connections dictionary. Hence, when the original client went to disconnect for real, there were no clients with that ID connected, according to the dictionary. This was a simple fix, ensuring that you can only decrease the number of connections with the ID if you have increased that number upon connection. The error in the second client with wrong password is now fixed, this was a simple check for wrong password client side, now the client is informed that the password was incorrect and to try again. The server also realises that the connected client did not have the correct credentials and refuses the connection. Now no errors occur when connecting 2 clients with same ID but wrong password.

What we disregarded:

- **Disregarded Error 1:** It's not possible to log in without any action field in the given JSON file
- **Why we disregarded it:** If the action field was present in the JSON file, with an empty list of actions, it would be accepted, allowing the user to log in, however, if the action field is not present, the JSON file would not follow the correct format, thus being rejected. Therefore, we allow a user to log into the server only if the action field is present (even if empty), but without an action field it wouldn't make sense to let them log in as the format would be incorrect.

- **Disregarded Error 2:** The same username can be registered without a warning
- **Why we disregarded it:** You have to input the same password to log in with the same client ID. Essentially, if two people have the same username, they need to have the same password, and this check was already implemented.

- **Disregarded Error 3:** The clients will stop working when they receive an incorrect data format.
- **Why we disregarded it:** This was by design, when an incorrect form is given, the error is given, and will need to be run again.

- **Disregarded Error 4:** Similar to disregarded error 2, we already check that the passwords match, and if not, log-ins are not allowed with the same ID.
- **Why we disregarded it:** For the reason explained in error 2, it was a method to ensure security of the same username and password.

- **Disregarded Error 5:** No anti-debugger to stop attackers from changing variable values etc.
- **Why we disregarded it:** Outside the scope of this course.

- **Disregarded Error 6:** Delay is not applied to the first action
- **Why we disregarded it:** The way in which we implemented the delay in our project is that we have it between each action, therefore, if there's only one action in total, there won't be any delay for it to be executed. The only time the delay will be noticed is when there's more than one action, where the delay will happen between the actions.