

User Manual

by Pierre-Jean Besnard & Louis Billaut

Summary	3
Server usage	3
Client usage	4

Summary

ChatHack project has been developed during our first year of Computer Science Master's degree. ChatHack allows to send messages and transfer files by using a server and clients based on TCP protocol. Purpose of this user manual is to explain how use this project and describes all functionalities it implements.

Server usage

IMPORTANT:

- Before launching server, make sure that the "censure.txt" file is in your current directory. The server needs this default censorship file to be launched.
- Before launching the server, please launch the executable jar "ServerMDP"

A default database server is available, you can launch it by using:

```
- java -jar ServerMDP.jar port passwordFile
```

port : an int which represents the port that the database server will use.

passwordFile : a string which represents the path to a login/password file

(You can use the default passwords file: passwords.txt)

The passwords file should be written like this:

login1\$password1

login2\$password2

...

loginN\$passwordN

To launch the server, you can use the ServerChatHack.jar file.

The Server can be used with 3 or 4 arguments and must be called like this:

```
java -jar SeverChatHack.jar arguments
```

Arguments:

- port dataBasePort dataBaseAddress censorshipPathFiles
- port dataBasePort dataBaseAddress

port : an int which represents the port that the server will use.

dataBasePort : an int which represents the port that the database is using.

dataBaseAddress : a string which represents the address that the database is using.

ensorshipPathFiles : a string which represents the path of the censorship file.

This censorship file is a .txt file and enumerates all words that will be censored by the server. It must be written like this :

word1

word2

word3

...

wordN

If the server is called without precise any censorshipPathFiles in arguments line, the default censorship file will be used by the server.

Client usage

To launch the client, you can use the ClientChatHack.jar file.

The client can be used with 4 or 5 arguments and must be called like this :

- *java -jar ClientChatHack.jar arguments*

Arguments:

- path login host port
- path login password host port

path : a string which represents the path where files will be launched and received.

login : a string which represents the client login.

password : a string which represents the password associated to a registered login.

(This login is registered in the database used by the server.)

host : a string which represents the address that the client will use.

port : an int which represents the port that the client will use.

When the client is launched, you can use different types of commands:

- **message** allows to write a message in the global chat, this message will be received by all other clients connected.
- **@login message** allows to write a private message to another user, the other user will first received a notification which explain that you want to create a private communication with him. The other user can accept or decline this asking. The private message will be received only by the other user.
- **/login filename** allows to send a file to another user, the other user will first received a notification which explain that you want to create a private communication with him. The other user can accept or decline this asking. The file will be received only by the other user.
- **+login** this command is available when a user received a private connection asking. It allows to accept the private connection. This user will then received the message send by the other user.
- **-login** this command is available when a user received a private connection asking. It allows to decline the private connection.
- **#login** allows to close a private connection with another user.

The client isn't using the censorship for private connections.