Project ”KTN-Chat”

There have been some changes to the application.

## Structure

Structurally, the app has been divided into two separate .jar files; one for the server and one for the client. This is so that both a server and a client can run on the same machine.

In addition to this, there are only 5 classes, not 6. The classes are:

* Server
  + Server.java
  + ClientHandler.java
* Client
  + Client.java
  + MessageParser.java
  + MessageReciever.java

As you see, the packages have been removed, and both file structures are implementing *json-simple-1.1.jar*, which is a simple JSON API for java. In addition to this, the GUI is now implemented directly in the Client and Server classes, and not in separate classes.

## Functionality

The Client now asks the user for both username and IP-address before it is able to log in. The port is hardcoded into both the server and client. Once these two fields have been submitted, the client creates a connection to the server and checks if the given username is available, and returns a message if not.

When a message is sent by the client, it is put into a JSON object, which carries the request and its content to the server. There, the server is parsing through the JSON object, and deciding what to do. The response is sent back to the client as a JSON object, which the client parses through and prints the message to the user.

The following is a list of commands the server understands:

* “help”: The server/client returns a text to tell the user what to do in order to connect to the server.
* “names”: the server updates the list of names represented at the right hand side of the GUI.

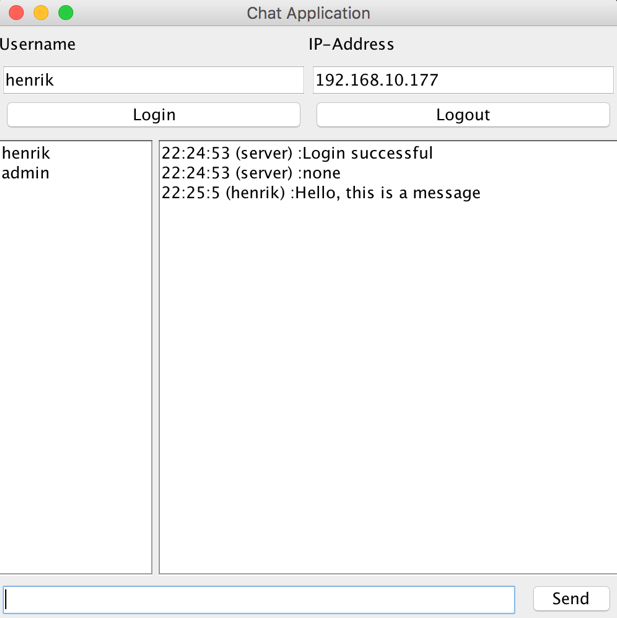
The following commands are built into buttons to improve the user experience:

* “login”: when the user presses the login button, the client sends a login prompt to the server.
* “logout”: the client sends a logout prompt to the server.

On the server side of the program, there are different responses for each command/request. The following is a list of possible responses:

* “help”/”info”: the server returns a text containing information on how to use the application.
* “login”: the server returns if the login was successful or of the username was taken.
* “logout”: the server is not returning any message, but the client is disconnected.
* “names”: the server updates the list of the currently connected users. Normally this happens automatically every time a client connects and disconnects.

## Graphical User Interface (GUI)



The GUI has been designed with ease of use in mind. In the center of the screen there is a window that shows the history of the chat. This is were all the action happens. On the right side, there is a list containing all the users currently in the chat room. On the top of the application, there is a textfield for username and one for the IP-address. Underneath these fields there are two buttons; one for login and one for logout.

At the bottom of the screen there is a textfield where the user can enter their message and a send button to send the message. One can also simply press the return key on the keyboard to send a message.

## Known bugs/errors

There are a few bugs that the development team knows of. These are being worked on, and should be fixed very soon. Below is a list of known bugs/errors:

* Client crashing when server is shut down
* Users not getting a message when the server is shutting down
* Client not being able to connect to the server through a firewall
* Users not being able to connect right after they have signed out

## Future functionality

As development continues, there are a couple of functions that are planned to be implemented. Some of these functions are:

* Encrypting all messages sent between client and server in a secure way.
* The ability to create several “chat-rooms”, and have several conversations going on.
* The ability to send messages directly to one specific user.
* The ability to create private users, stored on a server and protected with password.
* Color coding on users and server.
* The ability to send pictures.

Of course, these are plans for the future, and the main goal at this point is to keep the server running, and fixing all the bugs, so that the user has a smooth and easy chatting experience.