

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

12/17/2021

Design Document

Network Based Chat Application

Several thin, curved lines in shades of blue and grey originate from the bottom left and sweep upwards and to the right.

Máire Murphy
G00375722

Design & Implementation

A Client and a Server program called 'Natter' that use a reliable connection-oriented approach to communicate with each other using TCP Client and Server API via the java.net package.

Both Client and Server programs use a Configuration file called *config.txt*. The configuration file has three fields separated by spaces: host, port and thread poolsize. Current values are:

1. host name: localhost
2. port number: (65000 in the Dynamic\Private range). The server uses to listen on.
3. poolsize: 50 (number of threads allowed be used concurrently) .

ChatServer: On start-up , creates a server socket and run indefinitely accepting client socket requests. The server creates a thread for client tasks (no. of threads limited to specified poolsize). The *ClientHandler* thread ensures only users with unique usernames are tracked along with their socket in a *ConcurrentHashMap*. Then it listens for incoming messages from the client which it broadcasts to other connected clients. Housekeeping is performed on *ConcurrentHashMap* to remove users who are no longer connected.

ChatClient: Enables a user to be created with a username. It establishes a socket connection with the server. It contains two threads; one for performing a listening task (for incoming messages from other connected user) and a chat task (to send chat messages to the server). It allows the user to exit using '\q'.

Both programs use Console Colours. Broadcast messages from the server are shown in purple. Instructions to enable Console colours: <https://superuser.com/questions/413073/windows-console-with-ansi-colors-handling>

Instructions to Run

Unzip folder. **Note:** *config.txt* must be saved at same folder as *.jar file.

Option 1 Run Jar:

In ChatServer directory:

```
java -cp ./ChatServer.jar ie.gmit.dip.ServerRunner
```

In ChatClient directory:

```
java -cp ./ChatClient.jar ie.gmit.dip.ClientRunner
```

Option 2 Run Compiled Code:

To run code: In the *src* directory of each application, compile using: `javac ie/dip/gmit *.java`

ChatServer `java ie.gmit.dip.ServerRunner`

ChatClient: `java ie.gmit.dip.ClientRunner`

References Consulted:

- Java Docs:
<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/net/package-summary.html>
- Oracle Java Tutorials:
<https://docs.oracle.com/javase/tutorial/networking/sockets/index.html>
- Baeldung Tutorial: <https://www.baeldung.com/a-guide-to-java-sockets>
- Threading Examples from John Healy Java material and the course material provided in class.