Was ist bei **twmailer** zu tun?

**SEND**

Client:

* Zeilen nach der Reihe an den Socket schicken

Server:

* Zeilen vom Socket empfangen und in Datei schreiben
* Datei befindet sich um Unterordner des Empfängers
* Datei bekommt also Namen eine UUID
* LOCKING eventuell beim Erstellen des Empfänger Ordners erforderlich, da ja zwei gleichzeitig denselben Ordner erstellen könnten

Vielleicht die Datei erst in den Ordner geben wenn die gesamte Nachricht empfangen ist, sonst könnte ein LIST eine Nachricht ohne Betreff einlesen

**LIST**

Client:

* Zeilen nach der Reihe an den Socket schicken
* Zeilen nach der Reihe vom Socket empfangen
* Betreff anzeigen (runterzählen, weil man ja am Anfang die Anzahl bekommt?)

Server:

* Empfänger vom Socket einlesen
* Ordner öffnen und über alle Dateien iterieren, dabei die Subjects in einen Vector fetzen
* Inhalte vom Vector an den socket schicken
* LOCKING eventuell beim lesen der Datei erforderlich, weil ein anderer thread in der zeit die Datei löschen könnte

**READ**

Client:

* Zeilen an Socket schicken
* Zeilen vom Socket empfangen und anzeigen

Server:

* Username und Nummer vom Socket empfangen
* Ordner öffnen und bis zur nten Datei iterieren
* Datei öffnen und Inhalte an Socket schicken
* LOCKING beim Lesen der Datei, da ein anderer thread die Datei löschen könnte

**DEL**

Client:

* Username und Nummer an Socket schicken

Server:

* Username und Nummer vom Socket lesen
* Ordner öffnen und bis zur Datei iterieren
* Datei löschen, sofern diese nicht gelocked ist, andernfalls warten (was es eh macht glaub ich)

**QUIT**

Client:

* Befehl an Socket schicken
* Programm beenden

Server:

* Befehl von Socket erhalten
* Thread beenden

**Anmerkungen:**

* Vielleicht eine Bestätigung fürs Löschen fordern, weil wenn jemand LIST verwendet und dann eine Nachricht löschen will, dann kann es sein, dass jemand anderer eine Nachricht dazwischen gelöscht hat, was die Indizes verschieben würde.
* Allgemein problematisch ist diese Lösch Scheiße, weil wenn zwei dieselbe datei löschen wollen, dann lock ich den zweiten, aber er kommt immer noch dran und löscht dann halt was, was er vielleicht nicht löschen wollte, aber vielleicht wollte er es doch löschen