client and server architecture:

the client:

As specified the client can be started by providing an ip address and a port number to connect to the server. When connected the client handles all user inputs and creates strings in the correct format for the server to read. To handle user inputs the client first provides the user a menu with 5 options to choose from. Depending on the user input the client gathers follow up information (e.g. when sending an e-mail. It asks for a sender, receiver, subject and text). As soon as all necessary information is gathered the client creates a string in the correct format and sends it via the socket to the server. After that it waits for a message from the server. When it has received a message from the server it displays it for the user and then provides the user with a menu again.

the server:

The server accepts the port and spool-directory as arguments. The server waits for a connection from the client. Once a client is connected the server waits until it receives a message from the client. The server then processes the response and returns the appropriate response which is sent back to the client.

used technologies:

The server client communication is based on the TCP/IP and socket programming. Strings are mostly used for the message transfer between server and client.

development strategy and needed adaptations:

In the development we split the work between server and client communication and server and file handling. One programmer was responsible for the communication between the client and the server. Here it was necessary to create strings in the specified format, check for missinputs, change between sending and receiving data via the socket and extract information from the strings to then handover the correct information to the file handling functions.

the file handling functions read, write and create files in the corresponding spool directory. Furthermore, they send the messages back to the client with either the requested information or error messages.

usage:

```
tw_client <ip-adresse> <port>
tw_server  <port>  <spool-directory>
```