Data Models
Client :

- Login   Ex. LGIN user pass
- Logout   Ex. LGOT
- Store   Ex. STOR file.txt : file_content
- Retrieve   Ex. RETR file.txt
- Delete   Ex. DELE file.txt
- Menu   Ex. MENU
- List   Ex. LIST ↳ Requests commands from middleware

user
String

file
name
contents

Middle Ware :

- Same commands as client, but different protocol
- STOR, RETR, & DELE communicate w/ client + Db servers
- LIST only communicates w/ client after checking local user db.

file
hash + filename
file_contents

DB server:

```
┌─────────────────┐
│ files           │
├─────────────────┤
│ hash + filenam  │
├─────────────────┤
│ file_contents   │
└─────────────────┘
```

- Only needs STOR, DELE,
  + RETR protocol
- Needs protocol for
  overwriting existing
  file

* Mod_time

* Crea_time

* etc...

App Protocol:

Client -

Client Protocol (socket) → sends message
    ↳ listen (socket) → waits for response

Middleware -

MainProtocol (clisock, dbsock) → receive from client
    ↳ Decode (cmd, data, sock1, sock2) →
                decide what
                command

login (data, sock1, sock2)
    ↳ check if user exists, or fail
login_success (data, sock1, sock2)
    ↳ set globals + allow for more cmds

logout (data, clock1, sock2)
    ↳ reset globals + block commands

Menu (data, sock1, soc2)
    ↳ send menu to client

Middleware Cont. -

Stor_file (data, sock1, sock2)
        ⮡ send to DB
           If ok:
                update local DB
                send ok to client
           Else
                send error to client

Retr_file (data, sock1, sock2)
        ⮡ If in local DB:
              send req. to DB server
              If ok:
                  send. to client
              Else
                  send_error
          Else :
              send file_DNE error

DELE_file (Data, sock1, sock2)
        ⮡ If in local DB:
              send_req to db server
              if okok:
                update local DB
                send ok to client
              else :
                send_error
          Else :
              send File DNE error

list_file ( ) :
   get_files from local DB
   If None:
      send None to client
   Else :
      send fi+er to client

App Protocol Cont:

DB Server:

db_Protocol (sock1, sock2)
  ↳ Receive from middleware
     Then decode

Decode (data, sock1, sock2)
  ↳
     call protocol or send_error
     if cmd DNE

Dele ( )

Stor ( )
  check_overwrite
  overwrite

Retr ( )

Decode_file_contents (string)
  returns file_name, file_content.

Still unsure depending on how & where we store on DBs 1-4