KeyWave

Plan

- Initial fiecare scrie o parte din server (I.1 - I.3)
- Fiecare isi scrie cod de client pentru functionalitatea de server pe care o scrie
- Ideal ar fi sa folosim ceva pachet de testing

Todo's

I. Server

- server will be `async` (through a package named `asyncio`, or something along that line)

I.1 Database - store user/pass pairs

- some `sql` package (`sqlite` probably)
- verify if login info is correct
- add / delete users

I.2 Map to store user/socket pairs

- hashmap / lru / something
- point is to have a get_socket(user) function to figure out where to send a message

I.3 Temp storage for messages

- some data structure to hold messages, possibly using `queue`s.
- if message recipient is not online (not in the map), store the message and send to the user at a later time

I.4 Message handler (use all of the above)

II. Client

II.0 Read connections file

- skippable until we finish the rest of the server
-

II.1 Login request on start

- prompt for user/pass
- socket connection request

II.2 CLI

- navigate between different menus with keys (like `Home`, `Chat with [name]`, `New connection`)

- updates handled by `system("clear")`.
- update on NEW_MSG event
- send messages to the server

III. Key exchanges

- (we'll get there a bit later)
- figure out how the protocol works
- handle key exchange events in the server
- store the newly made connection in the connections file for later usage

IV. Send encrypted messages

- encrypt with a specific key for the desired recipient
- send encrypted message to the server
- handle encrypted message forwards in the server

V. Documentatie

- asta o lasam la final, facem toti (si alte glume bune pe care ti le spui cand te-a parasit nevasta)