%logo%

|  |  |
| --- | --- |
| date | update |
| 30.10.2024 | First version |
| 4.11.2024 | Tables added |
| 1.11.2024 | Reworked |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1.1 – Intoduction

The purpose of this document is to present the concept of my project, a chatting platform for rare languages speakers called "InterHolon".

InterHolon is a chat platform for none common languages native speakers and language learners. It’s main feature is a system called “The tree of the society”, which provides us with a possibility to analize people’s language sets and help them practicing their languages, no matter how rare they are and also practice interlangual speech to understand their languages better. The explaining of all of these concepts will be provided down the text.

Other auditory of InterHolon are language learners (I. G. Duolingo fans), who want to chat, learning languages in parallel. We provide an in-built system that shows you mistakes you’ve done in your messages, which helps optimize learning. Also we have no images neither videos supported to not get users distracted.

Interlangual speech – a conversation between two people in which they use two different but usually connected languages to speak. Helps us to understand our languages better.

The tree of the society is a system built on the concept of the language forest, which structures all the language of the world into over a hundred of language families presented with so-called trees. It helps us recovering language connections.

1.2 – System overview

The app is a chatting platform for native speakers of none common languages and linguistics involved people. It’s design is inspired by modern chatting platforms such as telegram, whats app and signal. The application includes screens of menu and chatting as well as registration, login and settings pages.

An algorithm ran on the server responds on the task of finding for a user a proper collocutor to practice interlanguagical speech.

Admin has access to a special window of administration

1.3 – Operating systems, Software and end-user enviroment

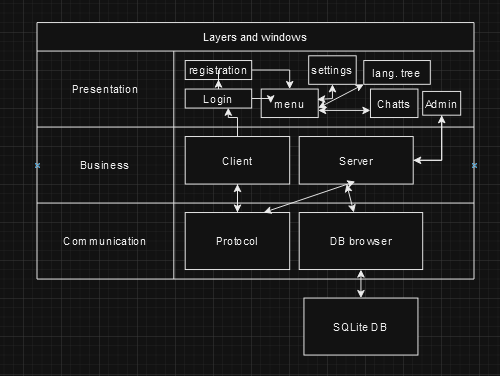
My app is ran on windows operated systems.

It needs no special knowledge but reading this documentation

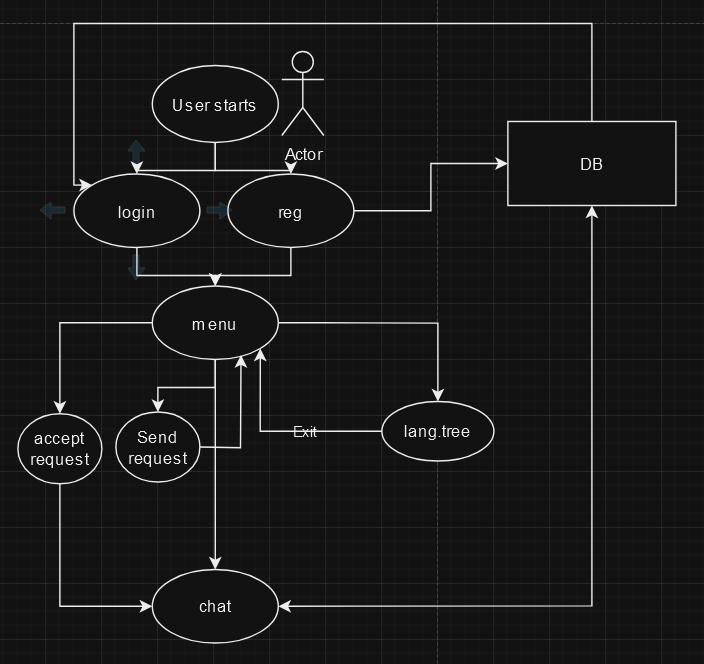
1.4 – blank space

1.5 – System architecture

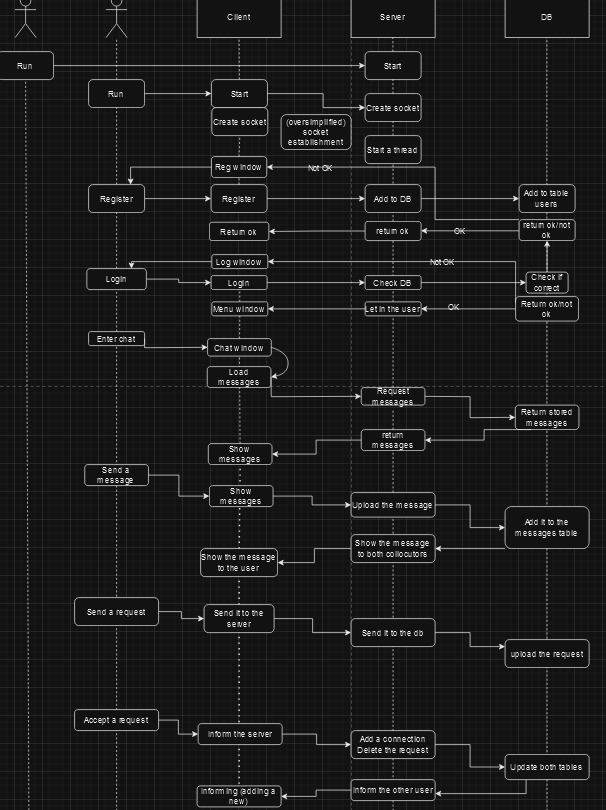
My system contains a bunch of windows separated into three layers



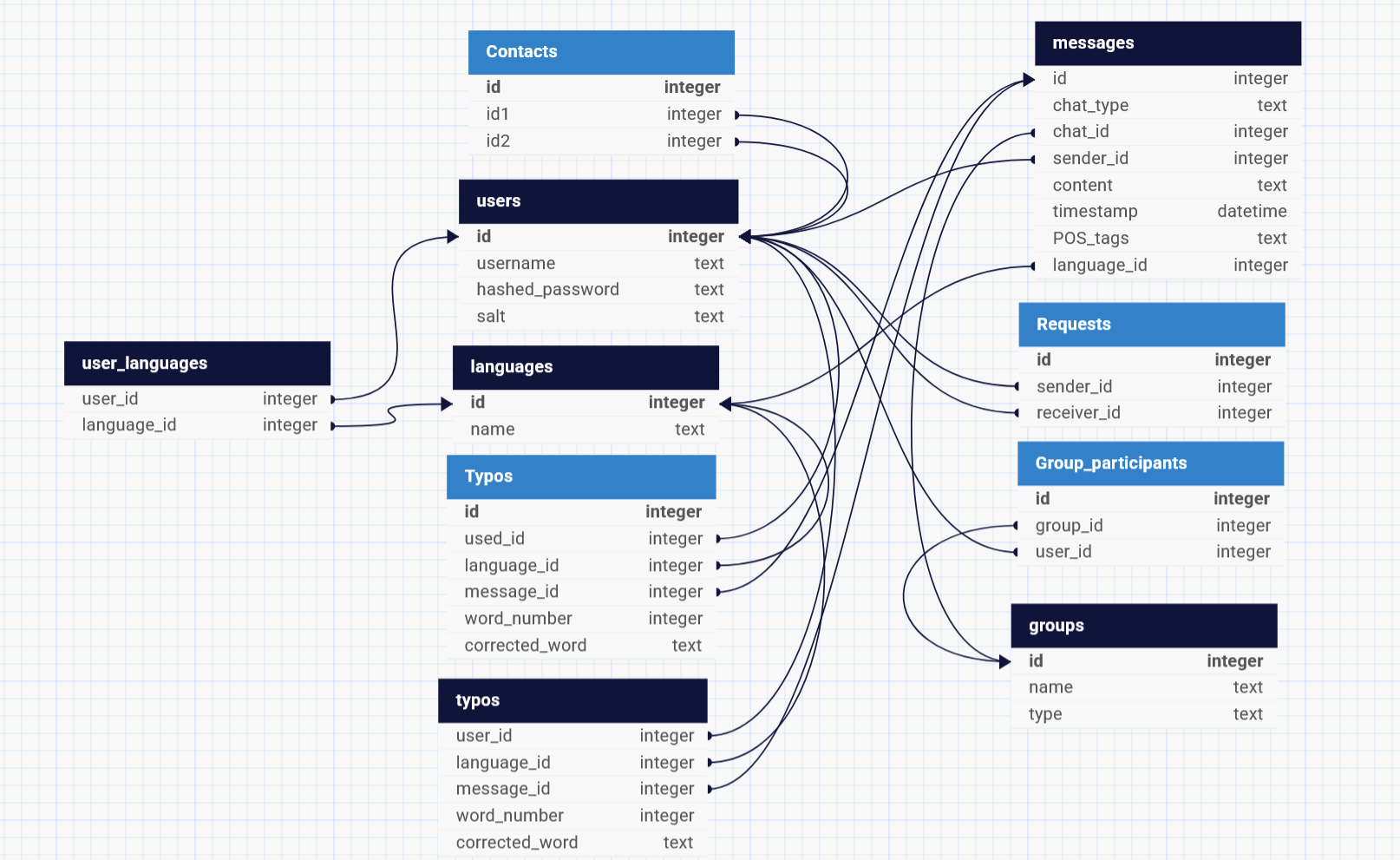
Also I prepared a user-case diagram showing the expected experience of a user



And a sequence diagram showing the correlation between layers



And also I have a scheme of all the data tables I need and their connections

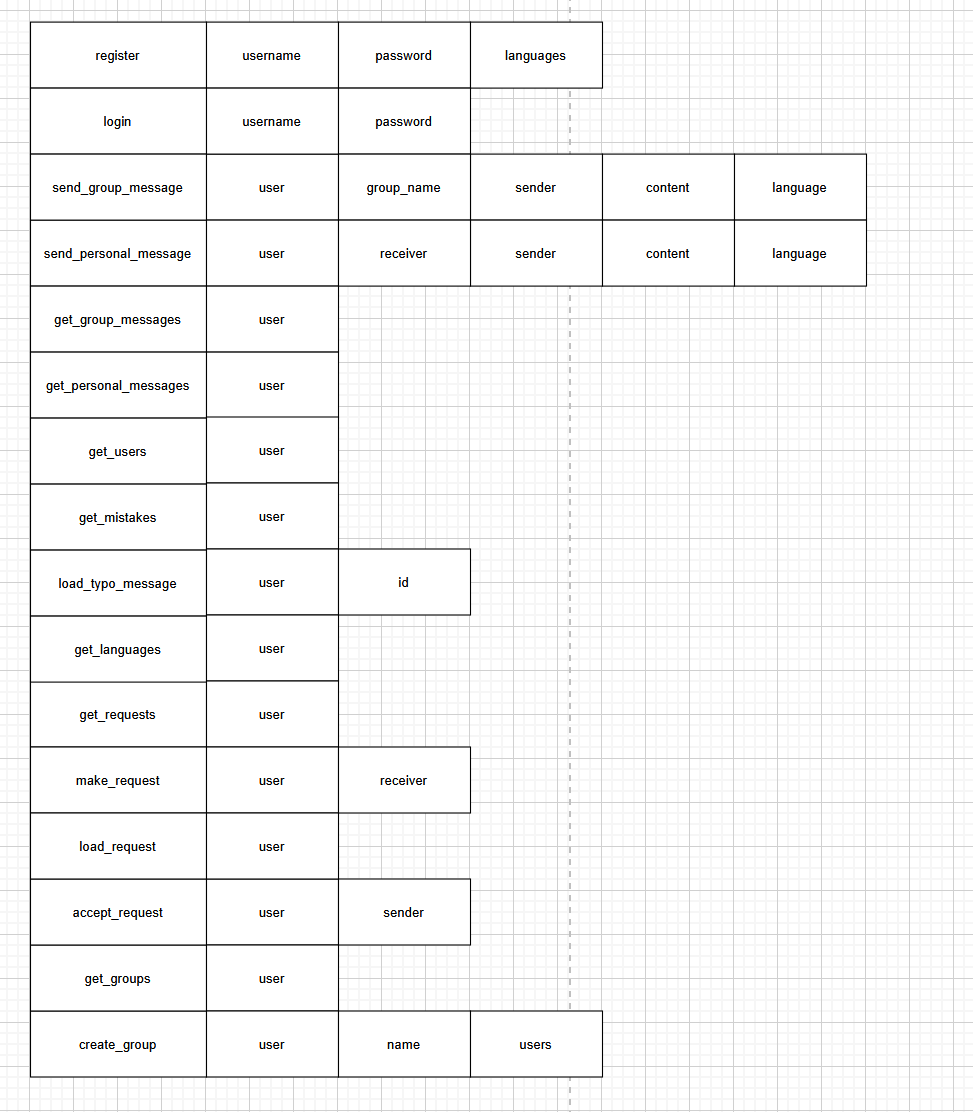
And also I prepared a blank version of a protocol for sockets

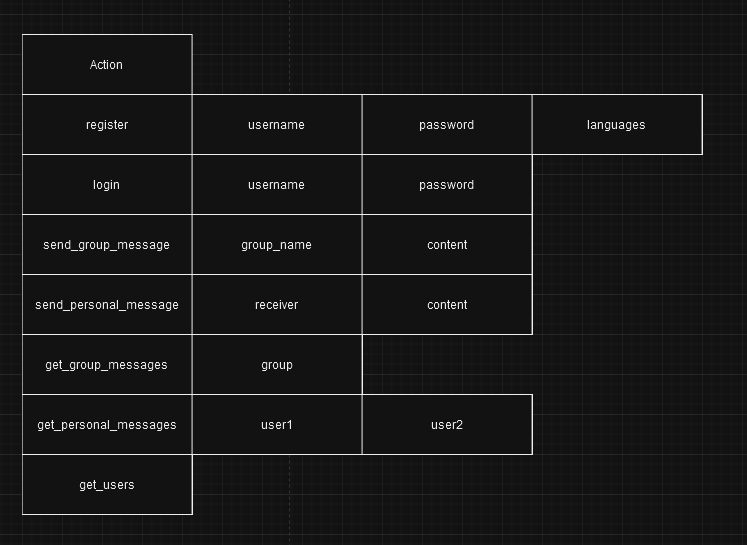
1.5.3 Protocol

Protocol:

I use JSONs (json.dumps(dict), to be exact) to transport data between server and client. The dict always includes “action” key, value of which provides type of the query. Other keys of the dict are specified for each request separately

For each possible request I listed needed arguments and expected responds:



The current scheme looks this way

1.5.4 passwords are stored only hashed with scrypt algorithm and salted. Not yet implemented other encryption

1.5.5 classes are Server and Client so far. In the next update gonna separate into BL and GUI both of them

Workflow

