Тестовое задание

Ссылка на репозиторий

Основная идея

Я решил сделать чат на основе запросов и ответов. Причем, чтобы эта система была децентрализованной я сделал чтобы каждый пользователь имел **клиентскую часть** с помощью которой отправлял сообщения и команды на серверную часть других пользователей. А также **серверную чать** на которую приходят запросы от других пользователей.

Особенности реализации и проекта в целом

- Когда приходит новый запрос считываются данные которые прислал другой пользователь и закрывается соединение. Далее открывается новый поток в который и передается запрос для обработки.
- Каждое зашифрованное имеет цифровую подсись сообщение проверяется когда его получил другой пользователь. Если подпись неверная, то сообщение отбрасывается. Публичный ключ запрашивается у адреса отправителя
- Админы могут выгонять (как на время так и отключать) и выдавать права новым админам. Ничем они больше не отличаются

Смайлики и другие символы

Чат использует кодировку utf-8 и будет возможно передавать очень много видов текстовых смайликов. В том числе эмоджи и текстовые смайлики

Отказоустойчивость & Децентрализация

Каждый пользователь хранит информацию о чатах в которых он состоит, админов, пользователей которые забанены, и остальных пользователях. Таким образом если даже выйдут админы или пользователи сообщение в чате получат те кто онлайн.

У чатов есть просто имена, а также *chat_id* - уникальный (для списка чатов у каждого пользователя) набор символов по которому в основном определяется в какой именно чат отправляется сообщение.

Проблема с chat_id

Понятное дело, что в децентрализованной системе некому будет следить за тем чтобы все *chat_id* были различными, так что может сложиться ситуация что будут чаты с одинаковыми *chat_id* и пользователь захочет зайти в оба у него не полуичтся потому что *chat_id* должны быть уникальными. Для этого есть возможность у чата сменить chat_id. Но у этого есть и обратная сторона, если злоумышленники зайдут в чат (* в него можно зайти только если админ подтвердит) могут зафлудить сообщениями о смене *chat_id* и из-зв того что могут быть задержки или вовсе потерятся пакеты некоторые пользователи могут потерять *chat_id* что будет означать что они больше не смогут отправлять и получать сообщения. Конечно, от этого можно попробовать сделать защиту. Например сделать новый вид запросов для **потерявшихся** пользователей и отправлять им. Или сохранять первоначальный chat_id и писать туда в случае неполадок. Но я решил дать выбор админу за тем что чат будет менять *chat_id* или нет.

Шифрование сообщений

Есть сообщения которые **не шифруются**. К таким относятся запрос публичного ключа, отправка публичного ключа, сообщения об ошибках и сообщения которые не шифруются.

Также есть сообщения которые **шифруются.** Но лишь отчасти. Сейчас понимаю что можно было шифровать и все сообщение, но на предыдущих этапах разработки я решил шифровать лишь данные сообщения. А тип

сообщения, имя отправителя, и в некоторых сообещниях *chat_id* передаются в нешифрованном виде

Чтобы значительно улучшить этот чат было бы хорошо добавить историю сообщений которую можно будет запрашивать у других учасников чата с определенного момента (например после отключения). Для этого уже реализовано сохранение сообщений.

Также сделать GUI. Но это очень трудоемкое занятие, а раз это тестовое задание не считаю необходимым этого реализовывать.