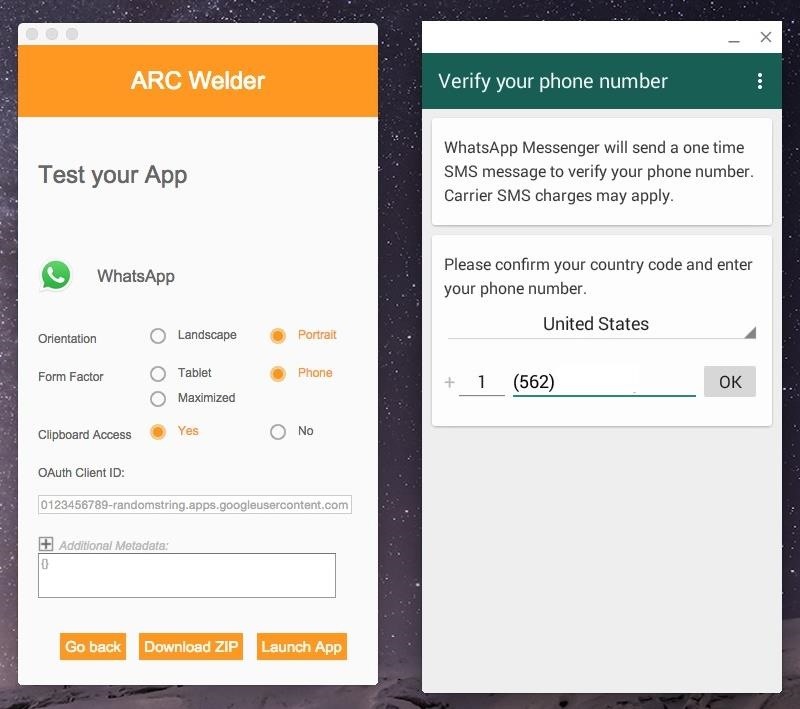
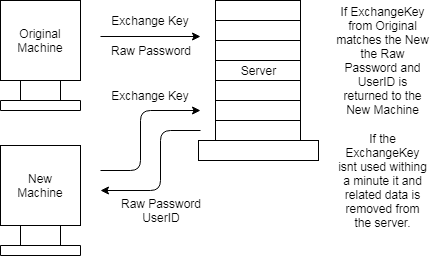
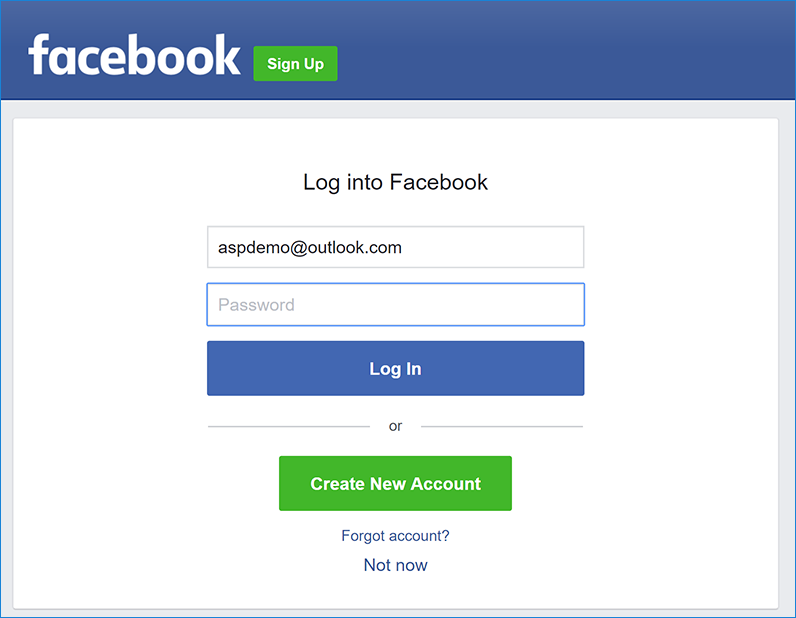
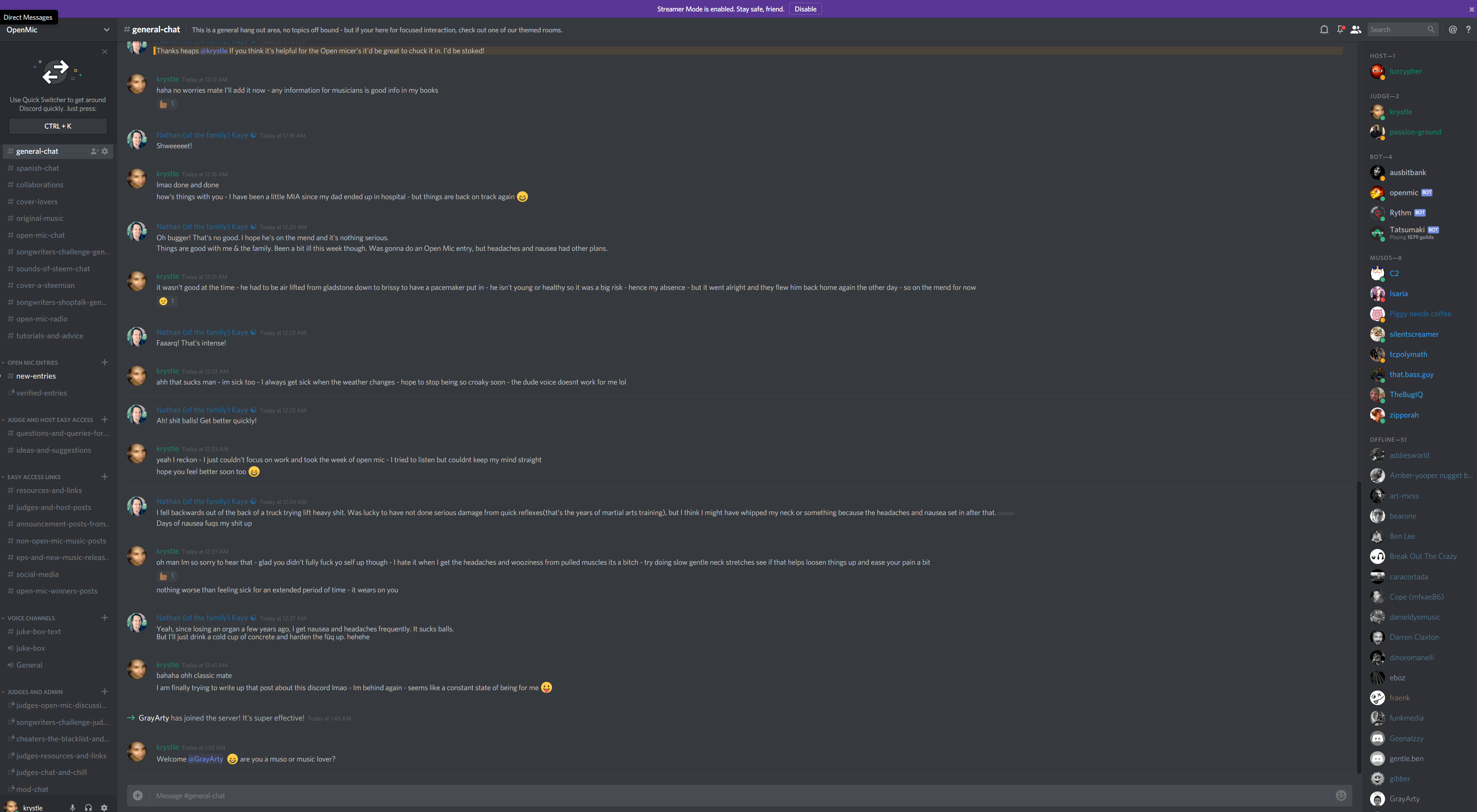
Background

I intend to create a P2S Chat system, which will allow for users of low technological ability, or those who wish for a simple experience, to textually communicate with anyone inside of the Rooms they are part of. In addition, I aim to combine some of the better ideas from other chat services, to create a superior option.

The best Chat system I know of (in terms of simplicity) is WhatsApp, it allows the user to easily create an account using their phone number and then begin chatting to Contacts already present on their phone. Which is great, however there are no Chat programs as simple as WhatsApp on PC.  
Most on PC require the creation of an Account attached to an email and an associated password.

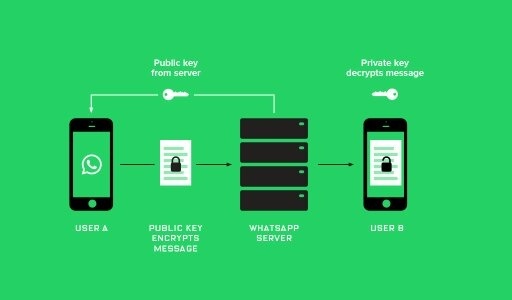
Hence, I intend to require only 1 piece of information on setup, their desired nickname.  
Then a random password and the unique user ID will be generated and stored on the machine.  
These will then be used in the future to sign the user back in.  
  
However, this will mean that on other devices they would not be signed into the same account.  
Therefore, I will add a system where a 6 character Key can be generated on the original machine and then entered onto the new device. An option will then appear on the original device asking for the input of a second code generated on the new device, if the code on the new device is entered correctly on the original device the UserId and password will be transferred to the new device and stored. Allowing for access on the new machine.

A fault with this system is that, in the circumstance access to all devices signed into the account is lost; it will be impossible to regain access to the account.  
Hence, the option to attach and set a personalised email and password will be available, turning it into a more Facebook messenger like system. Which will disable the ability to perform the Key Exchange sign in. However, will allow the user to access the account without needing an already signed in device.

I also intend to improve on WhatsApp group feature, by allowing the creation of Rooms/Groups, which then contains multiple Channels, allowing different discussions to take place simultaneously without getting in the way of each other; this is based from Discords room channel structure.  
This will also add the ability for channels to be created that only Moderators and the Owner may see.

Like most room based chat programs, I will allow the owner of the room to assign the role of moderator to members. This will then give the member the ability to remove members and messages , along with the ability to create and remove Channels in the room.  
However, they will not be able to assign moderator to other members.

Of course the feature to direct message a user will be present. However this will just involve the automated creation of a Room containing 1 Channel where no one is owner, and instead both are normal users. This will mean if they wish a new member may be added they can, but neither can remove or make moderator.  
These DM Rooms will not be convertible into a normal Room and hence only the 1 Channel will be present as no one will be able to create a new one.

However, unlike Whatsapp, messages will not be encrypted on the database. The only time any data will be encrypted, is during transit. Where the data will be encrypted using \_\_\_\_\_, where the Symmetric key will be exchanged using the Diffie-Helman key exchange protocol. Which should allow for relatively secure P2S encryption. But will be liable to Man-In-The-Middle attacks if it is performed during the key exchange.

Objectives

Client

1. Perform Diffie-Helman Key exchange when the connection to the server is started.
2. Send GET requests, using Headers to hold information.
3. Receive responses to requests over TCP.
4. Encrypt appropriate Header data using \_\_\_\_ symmetric key encryption method, to keep data secure in transit.
5. Perform User sign-in data exchange
   1. Create an Exchange Key and send it to the server along with the user’s login details.
   2. Send a given Exchange Key to the server, then receive and store the returned user’s login details.
6. User Initiated events
   1. Creation and control of user accounts
      1. Create a Password and Username less account, where sign in details are stored on the local machine.
      2. Ability to change Username, Password and other user data after the fact by sending a request.
   2. Creation and control of rooms
      1. Create a room with a set name and a source channel.
      2. Create and delete channels inside of rooms.
      3. Add and remove users from a room.
      4. Assign Moderator to users in a channel.
   3. Send messages in a channel.
   4. Sign-in using user provided details, or using stored details.
   5. Sign-up
      1. Using only a user provided nickname and create a password, get the userid and store the details on the user machine.
      2. Using a user provided email, password and nickname

Server

1. Receive GET Requests via a HTTPListener, that will allow for clients to send and receive data, via TCP. Without the need for the Client to have an open port in their network.
2. Perform the Diffie-Helman Key exchange when a new client connects.
3. Decrypt and Re-encrypt data using the \_\_\_\_\_ symmetric key encryption method, in order to keep client data secure during transit.
4. Respond quickly, within 1 second, using JSONs to carry formatted data, that will inform the client if the request was successful along with any error message or response message from the action.
5. Use Number based instruction identifiers, to reduce the likelihood of the action being misunderstood.
6. Interact with the database in order to read, update, create and delete relevant data from the tables. Allowing for user requested actions take place.