User Story 4 – Task 17- How the Server-side Works

The server for the application utilizes the Express framework for Node.js. The server stays running through the use of a docker container. The server uses the ‘sqlite3’ plugin to build the database that stores the information about the chat rooms and the users that are connected to those chat rooms. The server communicates with the application through sockets. In the current configuration of the server, the client/user communicates with the server through socket emitters to send data:

* join – gets the users name and adds the user into the chat room that they are trying to join
* getRoomInfo – gets all the users that are in the specified chat room
* exit – removes the user’s name from the chat room database
* disconnect – disconnects the client from the socket
* streamingToServer – sends the users voice data to the server
* endServerStream – ends the users voice stream to the server

There are also some functions to handle private chat rooms:

* joinPrivate – puts user into private chat room
* privateInit – sends a list of all the users invited to the private chat room
* privateStreaming – streams user’s voice to the server and other users in the private chat room
* privateEndStream – ends the user’s voice stream connection to the server

The application uses these emitter listeners to receive and process data emitted from the server:

* invitedPrivate
* privateStreaming
* privateEnded
* ready
* streaming
* joinedRoom
* leftRoom
* roomMembers
* endStream