**SIGNALLING**

During the life cycle of this application, from the time of login to logout, different components of the application will need to inform others about changes, request other components for their services, and provide feedback (and data) for a service provided (to another component). This section will describe the general messaging protocols/format that will be used by these components.

WebsocketWorkerThread: This is a dedicated web worker responsible for managing the web socket connection. It has the following responsibilities;

* Creates a web socket and establishes connection to the server side messenger applications.
* Ensure that the socket connection is alive through the life of the application by re-establishing connection, if connection fails while user is still logged in.
* Letting the user know when the server cannot be reached due to a network disconnection. When this happens it will send a "**NETWORK\_UNRACHABLE”** message to the main thread. Also receives user messages forwarded from the server and send them to the main thread and forward it to the server: "**CONTACT\_MESSAGE”.**
* Receives user’s online status updates and request and forward them to the server: **STATUS\_REQUEST, STATUS\_UPDATE.** Also receives status request and updates reply from the server and forward them to the main thread.

**MESSAGE NOTIFICATION QUEUES**

**QUEUES MESSAGE/NOTIFICATIONS**

* If a user sends a message and the socket is still in CONNECTION MODE, the message should be placed in a queue and send when there socket has connected to the SERVER.
* When a user receives a new message, the message should be QUEUED if the context of the sender is not opened. The message notification should be displayed (and incremented for each new message). When the user enters the context of the sender (by clicking on the sender’s name) the MESSAGE VIEW should be updated with the new messages (colored differently to distinguish them from old ones).

**NONE QUEUED MESSAGES**

* **Add Contact request notification**
* **Group-join request notification**