Units group:

You need to create the socket as such:

Client client = new Client("localhost",1234,"keystore.jks","password");

Client client = new Client("localhost",1234,inputStream,"password");

Before sending another command, you have to check if another command is being processed:

if(!client.getWorking().get()){ }

When you want to send a command you need to use:

client.setMessage("test_1");

Then you need to check if you received a reply for the command:

if(client.getFinished().get()){ }

And then you can get the command:

String reply = client.getReply();

When you want to check the queue, you can get it like this:

ACQueue ac = new ACQueue();

HashMap<Integer, String> queue = ac.getItems();

The following boolean can let you know if there are new commands in the queue:

if(ac.getHasAddedCommands().get()){ }

And once you have, you should set it to false:

ac.setHasAddedCommands(false);

Servers and DB group:

Communication with the client:

You need to get the queue list like this:

WriteQueue que = new WriteQueue(1);

HashMap<Integer, Item> items = que.returnMap();

Next you sort the commands that are not answered or old, put all the old commands at the end of the queue by setting their priority to 0:

```
if (items.get(i).getState() && !items.get(i).isAnswered()) {
        if (items.get(i).getUser()==null){ //IF LOGIN then SORT and PREPARE TO LOGIN, ELSE
                                                SKIP }
        else {
                Sort using items.get(i).getUserPrio() and the priority of a command. Also, here you
                should set the priority of the command in the item list so you don't have to check
                again: items.get(i).setPriority(int); and items.get(i).getPriority() to get it.
```

```
        }
}
```

Communication with the arduino:

Create the connector:

ArdConnector ac = new ArdConnector("PORTNAME");

ac.start();

When you want to send a command check if a command is already being processed:

if(!ac.getWorking()){ }

Then you need to set the command using:

ac.setCommand(String);

Then you have to check if the command has been answered:

if(ac.getFinished().get()){ }

Then to get the answer you need to do (NOTE: you can only get the answer ONCE):

String reply = ac.getInputLine();

Communication from the arduino to the clients, queue and multicasting:

To get the queue list:

ACQueue ac = new ACQueue();

HashMap<Integer, String> list = ac.getItems();

These items need to be put in the database and sent to the users of the house that are connected:

You can find which users are connected by using the instance of the server you created before:

HashMap<Integer, Communication> threads = server.getThreads();

Then you can get the user from each thread using:

String user = threads.get(i).getUser();

And then you can send messages to the user that you want using:

threads.get(i).sendUpdate(String update);

Arduino group:

Doing nothing:

You can get a chair using:

Your hands.

And then a laptop:

Your hands.

Open the web explorer of your choice to order a pizza:

Click it.

Type www.onlinepizza.se

Enter your information or mine.

Buy a pizza.

Wait for it.

Eat pizza.