**“Server-less” Java Chat System for Subnets**

For this product you will be designing a chat client that can be run on a local subnet, and will be able to discover and communicate with other clients without the need for an “official” server. The client will have a range of features including at the very least, group chat, rich text formatting, and peer to peer private communications.

**Project Goals:**

The following will be required

1. A “server-less” or “server delegate” system (ie. Client is chosen to take on server roles)
2. A group chat interface
3. Usernames (non-permanent, unique naming)
4. Private peer/peer “whisper” communications
5. Simple text formatting

Additionally, I would like to meet as many of these additional goals as possible:

1. Private and non-private sub-chatroom interfaces
2. Encrypted traffic
3. Locally hosted profile and other static information
4. Embedded in-chat images(jpeg/gif), videos(youtube embed), music (quicktime player embed), etc.
5. Peer to Peer file transfers (or other type of file share)
6. Rich text editing
7. Bulletin/MOTD/HTML greeting page
8. Modular plugin system
9. Anything else the coders can imagine!

This project is interesting mostly because the current chat systems either require a server or are designed for WAN, and require extensive configuration and a dedicated server. The LAN messengers that are in fact server-less (ie. Fortran messenger) are outdated and have no dynamic features such text editing, username registration, image embedding, etc. Creating a general interface to share images, messages, videos and raw files within a home, or a dorm setting would be a fun and exciting tool for many college students and families.

The project is highly feasible given the ease of configuration of tcp sockets in the java libraries, and the stable, straightforward APIs available for ideas such as the HTML greeting page and image rendering. Additionally, given the fact that most everyone in the class has at least 1 year preliminary experience with java, it should be no problem to get started with message sending/receiving and most other general data parsing.