

KC Chat Project overview

Iain Moug

September 3, 2012

1 Description

The idea for a Kelvin Connect specific messaging system had been mooted by the Customer Delivery team on several occasions with the perceived annoying characteristics of *Windows Live Messenger* such as the advertisements, the friend system, the size limit of messages and the formatting of code which was felt greatly restricted the transmission of code on the platform.

2 Structure

The project is organised into two packages; client and server. Each of these packages are designed to be modular since both these applications are supposed to be run independently of each other excluding the networking aspect. There are two text files within the project folder which hold user and friend data as well as a folder which will hold Avatars. (Structure below correct as of September 3, 2012):

- **KCChat**
 - **client** - *The client package.*
 - * **C_BackEnd.java** - *The class which does all the processing and networking for the client communicating with the Server.*
 - * **C_GUI.java** - *The client's user interface.*
 - **server** - *The server package.*
 - * **S_BackEnd.java** - *The class which does all the processing and networking for the server as well as carrying out many of the requests sent from the client via the S_SocketThread class.*
 - * **S_GUI.java** - *The server's user interface.*
 - * **S_ListeningThread.java** - *Thread that listens for new connections and then calls S_BackEnd's listen method.*
 - * **S_SocketThread.java** - *Thread which listens on a given socket for requests and then calls the appropriate methods in S_BackEnd.*

- **Avatars** - *The folder for the Avatars, this is where the JFileChooser in the GUI points to.*
 - * **avatar1.png**
 - * **avatar2.png**
- **friends.txt** - *Plain text file which contains numerical data relating to all the user's friends. (May soon be deprecated)*
- **users.txt** - *Plain text file which contains user data including; nickname, local IP address, email address and password.*

3 Features

3.1 Alpha

- Easy and convenient communication between all users/employees through the removal of advertisements and the “adding” of friends within the company. The company list will be managed centrally in the Server by an *Administrator* account.
- The networking will be handled by the two “BackEnd” classes of the Client and Server using the Java Socket, DataInputStream and DataOutputStream classes.
- The provision of an adequate user interface for both Client and Server using Swing.

3.2 Beta

- Code blocks, a way to send formatted and unrestricted amounts of code via the messaging client so as to avoid code break up and the use of other forms of less convenient communication like email or placing the code in a network location. First version will just send a block of formatted text, the code, regardless of language, later versions could perhaps have the ability to *detect* certain languages like *Java* or *XML*.
- All user data including usernames, email address, local addresses and password shall be encrypted in the final version but as of the alpha build this will be stored in a text file with test data in the following format: Iain, 192.168.38.120, imoug@kelvinconnect, 123.
- Suggestion to automatically log users into the chat client, would involve writing files into a user's AppData folder.

3.3 Release

- The possible extension within the release version for personalisation with backgrounds and avatars both of which have been factored in the Swing

alpha version with a JLayeredPane with the ability to have multiple layers and a JButton to display an avatar.

- The implementation of multi-user (or group) conversations adapting existing structure of *requests*.

3.4 Future

- Once the core functionality has been reached the possibility of enhancing the user interfaces by using a better interface library. One which has been suggested is *SWT*
- File transfer service, not sure exactly how this would be implemented at the current time.

4 Links

I do most of work using Google Drive to sync my workspace folder between computers. On Michael's suggestion I have made a Google Code project page for KC Chat, I am trying to keep the subversion up to date as well. If you want access to the code on either websites please let me know. I have also put a download for Eclipse 4.2 Juno which I am using to develop this project as well as created a wiki page, mostly for my reference, for certain websites which I have found useful in understanding certain problems mostly to do with networking.

- Link to Google Code subversion:
<http://code.google.com/p/kcchat/source/checkout>
- Link to Google Drive KCChat folder:
<https://docs.google.com/folder/d/0BylC6iojRmIPWFNjQXNJdkxGc3M/edit>