CS 3110: Final Project Team Charter

Team Members:
Colin Sullivan (crs335)
Jasper Liang (jxl8)
Yuet Ming Leung (yl787)

Meeting Schedule:
Time:
Fridays 3-5pm

Location:
Upson Homework Boxes

We will meet as necessary outside of the weekly meeting.The team will communicate using Facebook messenger and all code will be shared through GitHub.

Proposal:
A very short statement of the core vision or key idea for your system.

We intend to make an instant messaging system. We will provide a client application and a server application. Our chat system will enable users to exchange messages that contain text, images, and files.

A short (no more than six items) bulleted list of the key features of the system.

- Communication between any number of users through chat rooms,
- Login and Registration interface for client identification
- Contains a panel-based tiling interface for different functionalities adjustable with keyboard shortcuts
- Allows the transfer of text, files. and images. Maybe emojis.
- Text messages are labelled with user names, timestamps, and read statuses
- Possibly encryption on the server side

A narrative description of the system you intend to build. Go into enough detail that, if your charter were given to another team, and that team were never allowed to talk to you, they would still understand more or less the functionality that you have in mind.

The system that we intend to build is an instant messaging system. It will have a server/client based implementation, allowing users to message any number of other users. The users will also be able to create chat rooms containing any number of users. We will implement a login and registration feature that will allow users to identify each other. We will possibly store the passwords as a salted hash for security reasons.

Our instant messaging system will allow users to send text, images, and files to each other. The text can be formatted with color, underline, italics, bold, etc. and may also contain emojis. Messages will be labeled with timestamps, as well as the user names. Messages will be tagged with statuses like “Sent” or “Read”.

We will make an interface for our instant messaging system that will contain panels that contain different functionality. For example, there will be a sidebar containing the user’s chat rooms, a main panel for the user’s current chat, a text box for user input, etc. These panels can be manipulated through keyboard shortcuts, allowing users to “focus” on a panel and manipulate them. This allows them to open up any number of chat panels and tile them.

Roadmap:

MS1 (alpha):

Satisfactory:

- Can send plaintext messages to a server
- Can receive plaintext messages from server
- Basic interface to see messages

Good:

- Some functional layout for interface (with textboxes and panels)

Excellent:

- Text formatting

MS2 (beta):

- Login/Registration interface
- Labelled messages
- Some sort of tiling layout interface
- Keyboard shortcuts

MS3 (release):

- Chatrooms
- Complete tiling layout
- Image and File transfer

Design Sketch:

Modules:

- Communication
    - Server
        - User management
        - Chat log storage
    - Client
    - Communication protocol
- Interface

        - Panels
        - Parsing keyboard input
        - Formatting text
        - Displaying images

Data:

- User/Password database
    - Stored in a text file/SQL database
    - Maybe transfer to a hashtable during runtime
- Chat rooms, chat logs (server side)
    - Stored in a text file/SQL database
    - Queried during runtime
- Chat history (client side)
    - Store as a list of custom chat types?
- Chat message
    - Variant/Tuple
    - Messages would contain timestamp, message, user
    - Files
    - Images

Libraries:

- ANSITerminal
- Unix
- Cryptohash

Testing:

Most of our testing will be done through the interface itself.

- Login/Registration interface
    - Making users
    - Logging in
- Sending messages/files/images
    - Sent correctly with correct formatting
    - Files should have the same md5 checksum?
    - Rendered images should be identical
- Group chats
    - Making groups
    - Sending group messages