# Eilia Keyhanee’s journal for “Wilia Chat”

|  |  |
| --- | --- |
| 29/05/2017  *Define Document* | At School:   * Examined code written by dingle * Spoke about base theory with william.   At Home:   * Made progress on the software definition document, spoke over teamspeak to coordinate effort.   To do next:   * Go through the document with Mr. Dingle to make sure the questions are answered the right way. |
| 30/05/2017  *Define Document* | At School:   * Asked Mr Dingle about the questions we did not know how to answer on the Software Define Document. * Completed most of the document, will complete the whole document at home, with williams.   At Home:   * The Software Definition Document has now been completed.   To do next:   * Get Mr.Dingle to look over the document * Begin storyboarding. |
| 31/05/2017 | At School:   * Made minor edits to the Define Document * Made main boxes and arrows in storyboard   At Home:   * Begun the Screen Design Document * Created every window * Ran into an issue where the work could not really be exported have to ask dingle tomorrow. * Started a mock program to see if I could transfer any data over the internet. * Got error that says that no such host could be found. Must ask Mr. Dingle tomorrow.   To Do Next:   * Go over all current theory with william and Mr. Dingle * Ask Mr.Dingle why the mock program does not find a host. |
| 1/06/2017 | At School:   * Mr Dingle was away today and I could not ask my questions. * Further investigated the mock program to find the problem. * I found out the problem was that I accidentally referenced the text block that said “IP:” rather than the text box that the user could enter an ip address into.   To Do Next:   * Go over theory with Mr.Dingle and ask him where to create a gantt chart. |
| 2/06/2017 | At School:   * Asked Mr. Dingle about the gantt chart and data directory.   At Home:   * Filled out the data dictionary with the main variables, such as the tcp clients, listeners, hashtable, and byte arrays. |
| 6/06/2017 | Took a break from planning the assessment in order to build pc.  At School:   * Completed the Gantt chart task (not a traditional gantt chart, rather used microsoft planner). * Gave one week for each specific task, eg Single Client Networking, User Interface etc... * Looked over the planner with william.   At Home:   * Began work on the project. * Created three main parts of the server * The Main Program that waits for clients to join and retrieves their user name. Then broadcasts the User name and activates the next part. * The next part is the Idle function, which waits for the user to send a message, and then broadcasts it. * The broadcast function retrieves every entry in the ClientList and sends the message to them. * Also created the code for the client, which begins by connecting to the IP the user input and sending them the username the user input. * When “Send” is clicked in the client, the string in the input box is sent to the server. (which is then broadcasted). * A thread is also used to constantly wait for a message being sent from the server. And then it pastes the received message to the display textbox. * Ran into an issue where the display textbox cannot be edited by the second thread. * Need to ask Mr.Dingle.   To Do Next:   * Ask Mr. Dingle how to change the display textbox with the new thread. |
| 7/06/2017 | At School:   * Mr. Dingle showed me how to edit an aspect that is not controlled by the current thread. A method called “Invoking”. * Used the new method to edit the textbox. * The main application is now complete. |
| 8/06/2017 | At School:   * Changed the separation text from “|” to various special characters, to avoid bugs further down the track. * Started commenting out code. Explaining what each section does. Still have to check over code to make sure everything is commented. |
| 21/06/2017 | At School:   * Did some research about timestamps * Ran into an issue where I can’t change the time value from the default. * Attempted to make the emoji button change the textbox in the client. |
| 8/07/2017 | After taking a break and slowing down. (most of the class has just finished doing the Definition Document) I decided to continue.  William completed the client UI before the holidays, I must now insert the code.  At Home:   * William had not done the Title window * Created Title Window * Completed the code for the Title Window. * Decided to make server wpf, as it could all be contained inside one program. * Coded server, with some alterations to make it WPF. * Coded Client * Ran into an issue where the client crashes upon loading, stops at the connect command. Might have something to do with the fact that I’m hosting the server and attempting to join it in the same program? |
| 9/07/2017 | At Home:   * Decided that in order to fix my previous issue, I best make the program launch the original console version of the server as planned, as creating a WPF version creates too many complications. * Made the program launch the server |
| 11/07/2017 | At Home:   * Deleted the WPF server. * Commented out the Client code. |
| 18/07/2017 | At School:   * Discovered how to do timestamps.   At Home:   * Implemented timestamps to all user messages |
| 19/07/2017 | At School:   * Changed the display to a richtextbox * Failed to change the font of the timestamp compared to the message   At home:   * Learned to change specific messages using runs * Successfully made timestamps of smaller size than message * Added profanity filter function, with a database of inappropriate words. When a message is being distributed by the server, it goes through a foreach loop of the list, and checks to see if the message contains the word. If it does, the message is replaced with: “THIS IS A CHRISTIAN SERVER” * Upon updating the server I re-merged the server and the client. |
| 20/07/2017 | At Home:   * Added more entries to the profanity filter |
| 21/07/2017 | At School:   * Added comments to the client code. |
| 31/07/2017 | At School:   * Added comments to the client * Updated the profanity filter to include a few easter eggs as well as a more efficient filter. |
| 2/08/2017 | At School:   * Added comments to the server. * Begun on adding a quit function on both the server and the client. |
| 4/08/2017 | At School:   * Completed the quit function using special messages that work as commands and are not broadcasted. * Completed the client side quit by sending a message to the server when the user quits, this message ends the infinite loop the server uses. Also broadcasts a message to all the other users saying that the user was disconnected. * Ran into an issue where I cannot do the same with the server, because console applications cannot use events. * Fixed the issue by checking connection everytime on the client side. If a connection is not received the client says disconnected. |
| 14/08/2017 | At School:   * Went through all the code with william. Explained everything. * Also added a function to not allow spaced usernames. For later functions we discussed that we will add (clientlist). |
| 15/08/2017 | At School:   * Added a client list function by checking the beginning of the joined messages, retrieving the username and adding it to the list. * TODO: add removing names from list when user disconnects. |
| 16/08/2017 | At School:   * Met with a variety of new issues today. * Realized that the client list does not work well at all. * The second person who joins does not see the other users. * The second person who joins sees usernames twice on each message * Names do not get removed from list. * NEED TO FIX EVERYTHING!!!! * HEEEEEELLLLLPPPPPP!!!!!!!!!   At Home:   * Fixed issue of username showing twice for second client thrice for third etc. as well as the server showing things more than once. * Realized that almost everything in the foreach loop that went through all the connected clients did not need to be there, only the sending message lines need to be there. The issue was that the message would not reset, and the code kept adding the username at the front of the message each time it looped. * Need to solve the issues with the client list. |
| 18/08/2017 | At School:   * Attempted to create a system that sends all of the current user names to a new user that joins. * Ran into an issue where my system works in debug mode but not normal mode. * Will look more into it at home. |
| 19/08/2017 | At Home:   * Figured out the issue had nothing to do with debug mode, or the server speed. * Fixed the issue, and finally the client list, by changing all the special code messages to simple connected or disconnected messages that users get. * Ran into an issue where user connected and disconnected messages don’t show up for user. * Fixed that issue, ran into ANOTHER issue where whenever someone joins it says every user has joined… eg:   If Eilia is already in the channel, and William joins, rather than telling Eilia that William has joined, and telling William Eilia has joined, William has joined, it tells Eilia that Eilia has joined again, and then William has joined.   * Fixed the issue, created a system that checks if the username is already joined client side. * The client list finally works. * Re-merged the new server with the client. |
| 21/08/2017 | At School:   * Created some new comments for the code, still need more comments. * Realized my code doesn’t work, * Fixed an issue where the disconnected messages would not show * Encountered another issue where messages do not show. |
| 22/08/2017 | At School:   * Finally fixed the whole program. * I have no idea how. |
| 23/08/2017 | At School:   * Made it so closing the program does the same thing as disconnecting * Added all the comments for the server and the client, for all of the new features, adjusted spaces and comments. |
| 24/08/2017 | At School:   * Completed the spacing and the comments in the server code. * Removed some redundant code. |
| 25/08/2017 | At Home:   * Created a tutorial on port forwarding using the methods in IDT. |
| 28/08/2017 | At School:   * Re-completed data dictionary.   At Home:   * Tested the program on two computers with two networks. * Testing successful !  **Project Complete** |
|  |  |