Alpha Sprint Report
`Jammed` Alpha Release (1.0)
3.20.15

John

I laid out the initial class structure for the program, to help divide up the tasks. I then completed the cryptographic classes and message class for the program. One of the crypto classes was depreciated almost immediately when we found out we could use SSL. Since it dealt with encrypting our message packets for transfer across the network. This killed a fair amount of time(~ >5 hours) since Network Crypto was rather large and almost finished by that time. After completion the message class "Request" was later divided into subclasses for ease of use for other members, but later decided that we may not need subclasses, final decision will be made in a later release and will kill that productivity when we revoke the subclasses. I also dealt with findbugs for the project, which with tracking the bugs down and dealing with them with whomever wrote the code took a while. It is hard to keep track but overall time spent must be near 30 hours. Without the loss of productivity on the depreciated code, I would have had lots of extra time to pick up extra classes. Hopefully most of the design stage is behind us so deleting large sections of code will not be necessary. Our group did have a lot of trouble finding time where we were all available, which made the planning and working together harder.

Marcos

My primary task for this sprint was to create the database class for the server application to use to manage files. The initial draft of the class only took about three hours, but refining the methods and their functionality added several more hours of work, for a total of about eight hours. I believe these activities were a valuable use of time, because this is the interface by which the server will manage user files and logs. This was the only task that was assigned to me, and I completed it. For future sprints I hope to be more involved in other aspects of the functionality, especially since this particular aspect of the project is more or less complete. What's left is to add concurrency for multithreaded support, and I hope to assist more with the networking and security aspect of the project for the beta and final release.

Megan

My primary tasks were to create the initial user interface for the client application and to write the main method that the client application would use. Both of these tasks were necessary to the completion of the project, as some kind of user interface is needed for the program to be functional in the intended way, and the client application is needed in order for component code pieces written by many people to work together. I spent possibly around 10 hours working on both of these tasks. In the future, I would like to get more functionality into both of these classes, as well as working more closely with Communications so that the classes work smoothly together. In particular, we should get an in-app enrollment, user log requesting, removal of passwords, changing of user keys, changing of server password, deletion of account, and a UI GUI working, although some of these features will require assistance from others.

Dan

My primary tasks were to create the server application's main method and deal with the application level abstraction of Java's SSL library, namely reducing them to a simple send/receive/accept/connect/close interface. The Communication class was done first, which took about 3 hours overall. Next I wrote server class Jelly as a state machine. This class uses a large portion of our overall classes, much like our client class Jammed. There is work to be done with Jelly, namely adding concurrency support, and user authentication. Communication still requires work regarding proper SSL certificate usage and authentication, as well as more efficient and extensive error reporting capabilities.

Productivity Analysis

For this spring, communication took much longer than expected. Getting to the point where the server and the client communicated was very difficult. For the most part we got to where we wanted to be for this milestone with a couple of exceptions. The first of which is having the server and client run on separate machines rather than local host, and the second is having proper SSL certificates rather than using secret keys for the connection. In future sprints we plan on starting earlier, and having better lines of communication between group members.