Prakhar Singh
UIN: 428006559

CSCE 438 MP2 Design Document

In this Tiny SNS implementation, we have a client and a server that communicate using protocol buffers. On the client side, we create a stub and use it to send a login request using the login service in the server. In order to do this, we pass a request that includes the clients username in that is used on the server side to add the client to a database for management purposes. The request service is referenced from our sns.proto file, and also includes an argument called 'arguments' used for the other services (in this case for login we only needed a username, so the other argument wasn't needed). Once the server processes the login, it returns a status used to see if the request worked properly.

Once a client is logged in, they may now use one of four functions: follow, unfollow, list, and timeline. This is where its relevant to talk about the server. The server is started using functions in the grpc serverBuilder library; a serverBuilder object is created and the addListeningPort() and RegisterService() functions are called in order to initialize the server. All that is needed to start the server is a port number (we already know the hostname in this case being localhost). The server contains a database of a custom ClientStruct type, which represents each client. The ClientStruct contains a username, vectors for followers and following, and a server reader/writer message stream. When a user follows another user, the vectors of each struct are updated, and these vectors are used for the list request, as the list request requires the copying of the vector into a repeated string argument. The same applies for unfollowing, as the user being unfollowed is removed from the users following vector using erase and the user is removed from the unfollowed users' followers. On top of this, a method called 'user_in_db' is used to retrieve a user by returning the users index in the database.

The last and probably most important thing to mention would be timeline. In the client, whenever a user types in 'TIMELINE' two threads are started, a reader and writer thread. Initially, the write thread sends a message to the server that says "Initial Login". This message is meant to prompt the server to send the clients latest 20 messages from the users they're following. This is done by keeping track of who each user follows and then writing all the messages they send into a file. Now, when the "Initial Login" message is interpreted, the server then retrieves these messages from said file and prints them on screen. Meanwhile, the read thread receives messages using the grpc Read() function and displays them using the built-in client.h function displayPostMessage().

An important thing to note is that some parts of my code are a bit rough around the edges. For example, the "Initial Login" message I mentioned earlier becomes a part of every users timeline, even though its supposed to be an initialization message that isn't displayed. Another minor issue is the fact that I have an extra newline between messages, which can make the user interface of the Tiny SNS look somewhat clunky. Other than those two issues, everything else seems to work just fine.