Ethan Cherian
830002546
CSCE 438-500

Machine Problem 2 Design Document

Client

The client begins by submitting a login request to the server and, upon its success, becomes a member of the server's system. With its newfound access to the stub, the client can place any FOLLOW, UNFOLLOW, LIST, or TIMELINE commands, and the server will handle most of the backend logic. For the (UN) FOLLOW requests, the client and server must communicate through a message buffer, which allows the server to return any error information to the client. For the LIST command, a message is used, but not as extensively, only for tracking any changes to followed and existing users.

The TIMELINE request necessitates a bit more work, however, as a separate thread must be maintained to continuously poll a new Reader/Writer stream, and send a message to the server whenever a new post is made.

Server

The server keeps track of all existing users, as well as all users currently logged in to the system. Each user entry tracks all accounts that follow it, accounts it follows, and the last 20 posts to their timeline (stored as a deque). When the server is terminated, this information is stored to a series of .txt files on the disk, one for each user and one for the server's metadata. It accomplishes this by catching any SIGINT, SIGTERM, SIGKILL, or SIGQUIT signals received in main and calling an appropriate function. Whenever the server is started, it checks for the existence of these files and reads them in if needed, setting necessary class variables for intended functionality.

The server listens on a particular port for incoming connections from clients, which are simply handled by making sure that the username isn't already taken by an active user, before updating necessary server metadata. From there, all communication is handled by the GRPC protocol buffer, which abstracts away the minutiae of working with sockets and threads. The defined interface is simple enough, allowing clients to submit FOLLOW, UNFOLLOW, LIST, and TIMELINE requests. Upon receiving a FOLLOW/UNFOLLOW request, the server communicates to the client via a message buffer, wherein information about the request's status is kept; for the other commands, no such buffer is directly utilized, though the LIST request does use a buffer to store up-to-date information about existing users.

For the TIMELINE request, the server first writes all contents of the corresponding timeline to the screen (in reverse chronological order, to get most recent posts first), then begins reading from the stream set up by the client. Whenever a message is received from the stream, it is added to the user's timeline, replacing the oldest message if more than 20 are present. When the stream is no longer available, we know the client has terminated, so we set the user as inactive and invalidate their side of the stream.