

## MP2: Tiny Social Network Service

150 points

Due: March 6th, 2023, at 11:59pm

### 1 Overview

The objective of this assignment is to exercise your knowledge about Google Protocol Buffers and gRPC by building a Tiny SNS, similar in concept with posting and receiving status updates on Facebook or Twitter. In this service, the following must be considered:

1. Each user(client) has his/her own “Timeline” The user is allowed to post to his/her own timeline. That is, whenever new user appears, the server needs to create new timeline for the new user.
2. A user is allowed to FOLLOW any other user (i.e., is allowed to post updates to other users’ timeline), after submitting the FOLLOW command. For example, when an user posts an update to his/her timeline, followers can see it on his/her timeline.
3. All timeline must be persistent, i.e., their contents must be stored in files in the local file system. The files must not be encrypted, for now.
4. A user operates in two modes: “command mode” and “timeline mode”. When the client program starts, it automatically starts in command mode. The commands the client can send are: FOLLOW, UNFOLLOW, LIST, TIMELINE
5. The LIST command retrieves from the server the list of existing users and the list of users who follow the current user.
6. The TIMELINE command switches a user from command mode to timeline mode and allows the user to post some updates to both his/her and followers’ timeline.
7. Once a user issues the TIMELINE command, it can not return to the command mode, and it will stay only in TIMELINE mode.

8. The “UNFOLLOW <username>” command removes the current user from the given user’s timeline.
9. The “FOLLOW <username>” command adds the current user to the given user’s timeline.
10. After submitting the FOLLOW command, the user sees only new posts. That is, the user can not see any posts that were posted before the user started to follow.
11. After submitting the TIMELINE command, the user sees the last 20 posts as follows: “username”, “posting time” and “actual posting.”
12. All communications will be using Google Protocol Buffers v3 and gRPC.
13. The client and server must be started in the following ways:

```
$ ./tsd <port_number>  
$ ./tsc <host_name> <port_number> <username>
```

## 2 What to Hand In

The running system will consist of the tiny SNS server (tsd) (and possibly other servers) and the tiny SNS client (tsc). As platform, you should use the provided virtual machine where Google Protocol Buffers v3 and gRPC are installed, and develop the program in C++. (See the document “Programming Guide HW2”)

### 2.1 Design

Before you start hacking away, write a design document. The result should be a system level design document, which you hand in along with the source code. Do not get carried away with it, but make sure it convinces the reader that you know how to attack the problem. List and describe the components of the system: Client/Server Program, and their interaction.

For this assignment, we will not stress test your code. As long as the code executes correctly for up to 10 clients, one client at a time updating, you will receive full credit.

## 2.2 Source code

Hand in the source code, comprising of a makefile, a file `tsd.cc`, a file `tsc.c` and any auxiliary files (for example Google protocol buffers and gRPC files). The code should be easy to read (read: well-commented!). The instructors reserve the right to deduct points for code that they considers undecipherable.

## 2.3 Grading criteria

The 150pts for this assignment are given as follows: 15pts for complete design document, 15pts for compilation, 120pts for 6 test cases (the test cases have different weights). Refer to provided 6 cases tests which cover most scenarios but these are slightly different with the test cases for grading.