Project-2 -- Min(n)i(e)-Face

27.10.2020

Overview

To design and implement a simple internet tool that mimics the functionalities and features of Facebook. Note: UI design is not our concern/focus, but adding GUI is left to the Groups discretion.

Goals

1. Design and implement a Minie-Face tool using any of the networking paradigms that allows you to support the necessary Facebook features. User Management, Data updates, Notifications, etc.

Specifications

2. Design the networking paradigm that allows you to support the following Facebook features:

   Basic Features:

   a. Any Client/user should be able to register and set up an account with Mini-Face.

   b. Should be able to Login, get the updates and Logout.

   c. Should be able to search registered users and add/delete friends.

   d. Should be able to determine the list of active/online friends and initiate a chat session with them.

   Advanced Features:

   e. Support User to upload posts and categorize the uploads as either public (accessible to any user), private (only to friend circle) and strictly private (even Friends should not be able to access the post, but should be able to do so upon explicit approval from the user.)

   f. Scale to a concurrent server that can handle several client requests. Then, iff possible, try to scale to multiple instances of servers.

   g. Read offline messages and track read/unread messages for each client. Also, able to re-visit and read the already read messages.

3. Supporting a minimal set of security features.

   a. Users should be able to authenticate with the server before trying to access any of the features.

   b. When a user is prompted for a Login password, the user input for the password should be obscured/masked.

4. Other Bonus features (Grace points):

a. Adding Friends should allow the user to browse through the friends list of the newly added friends also.
b. Wall or the Timeline feature support for any client who visits another friends profile.
c. Good attractive GUI is always a bonus! -- Question is where will you add it?

Note and Some Hints:

1. As much as possible, try to design the application logic in such a way that you keep the Client side of the program stateless as much as possible.
2. Client programs can be as simple as running a telnet program (learnt in tutorial).
3. Use the Mininet Topology to automate and create a large number of user base (Client nodes). Implement using a simple Mininet Topology where Client and server functionality can be run on different end hosts. Example: Use a Simple Tree topology with Depth=5 that can allow you to have up to a total of 32 leaf hosts and any additional hosts that you can attach to other intermediate switch nodes.

Milestones and Deliverables

I. Design (40%)

HLD: Document detailing the networking paradigm. Handled Use cases. LLD: Proposed structure of networking paradigm. Client and Server side of the logic. State management. Commands and Action management. Security aspects.

II. Implementation, Feature Testing and Validation (30%)

Share the Client/Server and Mininet startup scripts used to realize the features. Upload your Code or share git link, with README.MD and instructions to compile and run your project.

III. Showcase your work (30%)

Prepare a 10-minute presentation highlighting the key design aspects, major challenges, issues faced while designing/implementation/testing etc. In the case of groups, be specific about individual contributions. Detail on: What is the outcome of this project? Did it help you or not? What could have been done better? It would be good if you can record a quick 2-3 minutes video demonstrating the working feature set of your project.