

Comment Anywhere Progress Report

April 3, 2023 - Week 10

Luke Bates

On March 29, Karl began work on the user banning pipeline. He added the new “DomainsBannedFrom” attribute to the UserProfile server-client communication entity. A GetDomainBans query is used to populate this data. Karl created the DomainBans table in the database, which does not use a primary key. Instead, the table is queried using a dual key being user_id and banned_from. This prevents a single user account from having multiple, redundant bans to the same domain. The particular DomainBan entry is deleted if the user account becomes un-banned. The actual ban records are stored in a separate table.

The BanActions table keeps track of all bans and un-bans, either globally, or for specific domains. The record will not be added if it is redundant. For example, if a particular user was banned twice, the second ban would be ignored, and not added into the database. This is one of the checks done by the Store.AddDomainBan function.

Store.AddDomainBan is the function that creates the ban entry in the database. First, it assures that the user in question is not an admin or global moderator, as these account types cannot be banned. Next, it will check to see if the user is already banned. This is done by attempting to add the ban record. If the function returns “pq: duplicate key value”, this indicates that the ban record already exists, and will respond with an error message saying you cannot ban a user that is already banned.

DomainModeratorController.HandleCommandBan will carry out the user ban. It will call a function to either carry out a domain ban or global ban. It realizes conditions for admins, global moderators, domain moderators, members, and non-existing users. To reiterate, admins and global moderators cannot be banned. Also, banning a nonexistent user will return an error.

On the front-end, Karl implemented the Ban Users Section. This is intended to be accessed by domain moderators, global moderators, and admins. If the domain field is empty, a global ban is invoked. However, this feature is restricted to global moderators and admins.

Also, if the user goes into their settings, they can see a list of domains that they are banned from. If they are globally banned, they will not be able to login in general. Bans are updated in real-time, in which if a user is currently logged in and gets banned, they will automatically be logged out, and become unable to log back in. However, users may be un-banned, in which their privileges are resurrected.

Luke created an EventManager class to keep track of all active event listeners where need be. This feature has already been implemented by specific components for click event listeners. However, this class allows any component to create and utilize any type of event. This is required for objects that are frequently created and destroyed, such as individual comments or individual comment reports. The issue is that event listeners persist until they are manually destroyed. Therefore, if a copy isn't saved to destroy later, this will cause a memory leak. Luke also created various UI improvements for the navigation bar.