

# **Comment Anywhere Progress Report**

March 13, 2023 - Week 7

Luke Bates

On March 8, Karl implemented moderator support for the back-end. The first goal was to correctly populate the `DomainsModerating`, `IsDomainModeratr`, and `IsGlobalModerator` fields for the `UserProfile` server-response entity. This logic was implemented into the `GetCommUser()` function on the back-end. If you are not an admin, it would check if the user was a global moderator. If not, and if the user is detected to moderate at least one domain, they would be identified as a domain moderator, and the list of assigned domains would be included in this response. During this time, Frank began work on designing the `CafeModeratorWindow` for the front-end.

On March 11, Luke created the `CafeDom` namespace, which was intended for wrapping primitive button and input elements into containers, and applying a consistent style. An issue occurred while I was implementing a stylized text-input box. I intended for the container to direct input to the primitive input box when clicked. The issue was that the container was a separate element from the actual input box, and there was no predefined behavior to invoke upon a mouse click. The solution was to add a mouse click event to the container, which would focus input on the actual text input box. This would give the satisfying effect that the container is apart of the text box, when in reality, it is not.

On March 12, Karl added the feature to allow admins to assign other users as admins. The biggest issue was allowing this change to update automatically, such that the targeted user, possibly being logged in at the time, wouldn't need to log out and back in for the changes to take effect. The solution was implemented as the `ChangeMemberControllerToGlobalModController()` function. First, the function would check if the user is logged in, or rather, if a user controller for the specific user id was present on the server. It would then assure the user was not already a global moderator. If both tests pass, the old controller is removed, and a `GlobalModeratorController` instance is created, and

becomes associated with that user. Finally, a LoginResponse is created to automatically log the user in as a global moderator.

On the same day, Luke configured the fetcher and dispatcher to handle CommentReport communication entities. These communications were utilized when a global or domain moderator requested a list of reported comments for a specific domain. If a moderator were to select a particular domain to search under, and click a button to “retrieve reported comments”, a ViewCommentReports entity would be dispatched to the server containing the specified domain. The server would then respond with a CommentReports entity, which would be dispatched to the CafeModeratorWindow as a list of reported comments, in which the moderator could then take action against these comments. The server successfully responds with this list, however, the list is not yet rendered to the window, thus moderators cannot yet act upon reported comments.