ECE 448/528 – Application Software Design

HOMEWORK #5

SOLUTION

1. Consider our members application discussed in Lecture 20-24 *(12 points)*

i) After the user clicks the checkbox to add/remove a member to/from a group, we wait until the server backend replies to update the state and to update the view. What would happen if it takes a while for the server backend to reply and the user clicks the same checkbox multiple time impatiently?

If the delay is caused by the server being busy, then the multiple requests will make the server busier. If the delay is caused by the browser being busy, then the multiple replies will make the browser busier. In either cases, the delay can occur even more if the user clicks the checkbox again after replies come back and thus checks/unchecks it incorrectly.

ii) To address the potential issues in i), someone proposes to disable the checkbox somehow after the user clicks it and enables it again once the RESTful response to get all groups arrive. List all events involved to implement this solution.

There are still two events involved. The first is when the user clicks the checkbox. The second is when the reply for getting all groups arrive.

iii) The solution in ii) should still be implemented winthin the MVC pattern. Discuss the changes you need to make for model, view, and controller. (Note that you don't need to implement it but need to discuss what needs to be added/mofied)

The above idea can be implemented as the following:

- **Model: Adding a variable to the state to indicate a pending request not replied yet.**
- **View: If the variable is set, we can display something to indicate q request is pending.**
- **Controller: We set the variable when the user clicks the checkbox and unset it when the reply arrives**

2. For password authentication, if the web application sends the account name and password directly to the server backend, it is possible for a careless backend develop to log the password in plaintext. Discuss a possible alternative. *(3 points)*

Instead of sending the password directly, we can send the hash value (SHA256 as an example) of the password to the backend, the server will only be able to see the hash value of the password.

This method can be made more secure (but with an extra communication) if the server uses salt to protect hashed password. In such case, the application will need to communicate with the server to obtain the salt for the given username first, and then compute the hash of the password with the salt before sending the hash to the server.