**Security Issues Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| Security Issue | Java File | Possible Vulnerabilities | Solution |
| No restriction of page view depending on role | Sqlite.java  Frame.java  Login.java | Lower-level roles like client may have access to admin privileges such as deleting a user and locking them which is normally not within their features | Implement a condition in Sqlite.java to check current user’s role, implement a button lock when navigating to the main page based on the role of the user |
| Unregistered or invalid users being able to access the homepage | Sqlite.java | Allows unauthorized users to access the homepage and affect the features present there which can impact actual registered users | Implement a condition to check if user is registered and has the right credentials before letting them enter, otherwise display an error message |
| Users can brute force credentials as there is no restrictions to the number of failed attempts | Frame.java | Malicious people can attempt to brute force a user until they can enter the account and possibly harm it | Implement a counter where if it exceeds the number of allowed attempts lockout the user for a specified amount of time |

**Security Issue 1 : Role Validation Code Snippets**

Function: code retrieves username from the login field and checks the corresponding role, it then returns it to the login action function where it locks the user to their own homepage

A computer screen shot of code

Description automatically generated

**Figure 1.** Role validation to check user’s role

A screen shot of a computer program

Description automatically generated

**Figure 2.** Before logging in, checks if user has an existing role and redirect them to the correct page

**Security Issue 2 : Register / Login / User Validation Code Snippets**

Function: When user enters their login credentials, go through a function where SQL database retrieves password and checks it with the current password if it matches. If it matches, user proceeds to homepage otherwise an error message pops up.

A computer screen shot of a program code

Description automatically generated

**Figure 3.** Password validation that retrieves user’s password

A computer code on a dark background

Description automatically generated

**Figure 4.** Function that retrieves the username and password field from the login prompts

A screen shot of a computer program

Description automatically generated

**Figure 5.** Before logging in, checks if user entered the correct password

**Security Issue 3 : Brute Force Security Implementation**

Function: When user enters their login credentials, go through a function where SQL database retrieves password and checks it with the current password if it matches. If it matches, user proceeds to homepage otherwise a counter will be incremented up to a maximum of 5 counts. After 5 failed attempts, lock the user from logging in for 10 minutes and reset the counter after.

A screen shot of a computer program

Description automatically generated

**Figure 6.** When user has less than 5 failed attempts, checks if user entered the correct password and redirects them

A computer screen shot of text

Description automatically generated

**Figure 7.** After exceeding 5 failed attempts, lock the user for 10 times by comparing the time the user got locked out with the current system time. After 10 minutes, reset the counter to 0 to reset failed attempts.