|  |  |  |  |
| --- | --- | --- | --- |
| **Security Issue** | **Modified Java File** | **Vulnerabilities** | **Solution** |
| Login not working | Login.java and Frame.java | Anyone can use the program regardless if the data inputted is valid or not. | Check with the database if the username exists and if the password matches the account |
| Password characters can be seen when typed. | Login.java and Register.java | If someone was watching the screen of an account owner, they could see the password and possibly steal it | Replaced textfield with a passwordfield |
| When registering, the program doesn’t check if the username exists in the database. | Register.java and Frame.java | Someone could name themselves admin and pose as the admin even without its privileges | Check if the username already exists in the database. If it does, warn the user with an error. |
| All homepage elements can be seen by the user regardless of role. | Frame.java | Users could modify data or see sensitive information | Upon logging in, the program hides buttons that would access other homepages. |
| If a button that should normally be hidden is still visible, the user could access normally hidden content | Frame.java | Users could modify data or see sensitive information that would normally be hidden | Upon clicking a button that would change the panel, it first checks the user’s role and if they have access to that content |
| Upon logging out or creating an account, the previously typed data is not cleared | Login.java and Register.java | Users could steal the previously logged in account | Clear the inputs every login, register and logout. |
| Application does not log user activities | LogWrite.java, Main.java, Frame.java and SQLite.java | Users could be accessing or modifying content not assigned to their role without anyone knowing | User logins, logs out, registers or them accessing content panels gets logged into a .txt file. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |