# 

# Security Meeting

## User Login Attempts

Discussed the current state of our user login attempt counting. Currently, the security feature does not appear to be functional. We are going to check our two different login methods to ensure that both are counting invalid login attempts and continue our troubleshooting from there. We plan, once functional, to allow 5 login attempts before locking the user’s account. We have not yet decided on how long to lock the account for.

## Database Assertions

We have already implemented many data assertions in our database, but we still need to evaluate all the parameters we are taking in from the webpage to ensure that everything is being properly checked as it comes in, before it’s used. We plan to look through all of the database code and make a list of the unique parameters being passed in from the site, then ensure that each of them has an assertion to match. We also discussed the possibility of putting common assertions (that is, assertions used in more than one package in our codebase) into one common assertion package.

## Clearing of One-time Passcodes

Currently, one time passcodes being sent when a user forgets their password are not being cleared until use. If the user remembers their password and goes back to the login page, the passcode is not cleared. Proposed fix is to clear it in the login functions and add an expiry date column for the passcode. This date can be compared against current time to see whether or not the passcode should be accepted, valid or not.

## Special Characters in Passwords

Currently, passwords can not contain any spaces or special characters. We are validating the password through a regex validator, and it is entered into an ASP TextBox. The TextBox has [type=”password”] as an attribute, although I believe this to be an HTML only attribute. Suggested fix is to use the ASP attribute “TextMode” and set that to password. Also suggested to look into creating a CustomValidator and overriding the OnServerValidating event to explicitly define everything we want to validate the password.

## Manual Decryption of Messages (Shoulder Surfing Protection)

We are concerned about the manual decryption interfering with other Pigeon features, such as the dynamically updating messages. We are unsure of the correct implementation to try to go for with manual decryption and shoulder surfing protection. If the user decrypts messages by hovering or clicking on them, those messages would just be encrypted again when the page dynamically updates the messages and rebuilds the repeater. It seems we almost have to choose between shoulder surfing protection and real-time messages. This needs to be brought to the attention of the user immediately.