**­­Name:** Lucy Gardener  
Project URL: ws328700wad.remote.ac

**WAD TEST PLAN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Page/Section | Test Details | Expected Outcomes | RAG | Comments |
| Website UX | Website should be responsive and accessible to users, irrespective of devices. | Website should be accessible to all users and work on a range of devices (eg mobile) |  | Ref A1:  While the page can be responsive to be used on different devices, in one stage it does get stuck and not work very well. Either have to make the program bigger or smaller to work again. |
| Admin Login | Entire System must be protected by a login page. | When accessing platform users should be prompted with the login page. |  | The index page redirects to login page if no login details have been saved. |
|  | Login page must only allow authenticated users to log in. | Enter *incorrect* login details. Login page should not allow access and display a warning/error message. |  | Ref A2:  If incorrect login details are given, the program returns box to show that the details were incorrect |
|  | Login page must only allow authenticated users to log in. | Enter *correct* login details. Login page should accept details and redirect user to chat platform |  | Uses bycrypt to make sure the password is correct so no plaintext passwords. And only works with correct details. |
|  | Users should be able to log out and clear their session when all actions are complete. | Use should be able to logout. If a protected page is requested, users should be asked to login again. |  | There is a log out button which simply destroys the session, and then redirects to the home page, which will then redirect to login page. |
|  | All admin/manager pages should be protected by username and password. | The following pages should not be accessible without logging in:  Admin Home  Add/remove/view users |  | Each user has values for different privileges. Only users with a specific privilege can do some things |
|  | Multilevel Access | Two user access levels should be provided: Users should not be able to access managers/admin pages. |  | My web app checks to make sure that you are an admin to be able to do certain things |
| Chat interface and real time messaging | There should be a system that allows users to send and receive messages, this should be done in asynchronous techniques. | When one user messages into a room other users should be able to see |  | I use both, websockets and short polling as a back up, this allows this work perfectly. |
|  | Users should be able to navigate through rooms | When looking at their rooms a user should be able to click on the different tooms and they should open. |  | When you log in on my web app, all chat rooms you can access are on the left, on clicking them they open like expected. |
|  | Messages sent in each room should be visible to all the rooms occupants. | I expect that all message should be sent to the database, and as well as that sent through websockets so users should get them instantly |  | Ref A3:  Users are able to see all messages that have been sent into the group and they are able to see when they were. |
| Room Management | Should be able to create, edit and remove chat rooms | Certain users are able to create and remove groups, and all users can edit them. |  | Ref A4  While groups can be created certain users, my web app lacks the ability to be able to edit them or remove them. |
| Admin Controls | Admins should be able to create, edit and remove users | Admins should have the ability to be able to remove users, create them and edit some of their details as required. |  | Ref A5  Admins can create and remove users. However they cannot edit their profiles |
| Additional Functionality | User profiles can allow users to have a picture bio or status creating a more engaging user experience. | Any user should have the ability to change their profile details. |  | Ref A6  Users have the ability to add profile photos for their profile, however any other details cannot be changed |
|  | Any changes made in the program at any time such as removing groups or messages should be seen by every user | Any changes made should be updated straight away to any user |  | This works in some places, such as when a group is made every user sees, however it does not work for everything |

How we protect the program using User Authentication and Privilege escalation.

During the login page, the program utilises google ReCaptcha to ensure that all logins are correct, as well as that when loading a page, it checks what privledge the user that is logged in has, and will stop certain things loading that they do not have access to. This stops them being able to use tricks such as showing certain models to use that they should not have access to by never loading it to them in the first place. Some JS files are not ran for them either.

How the program protects against SQL injections and xss attacks

In my program, I have a specific function that queries the database. In this I take all the parameters that want to be put into the database, and everyone is sanitised to ensure that they do not allow for SQL injections. This also helps protect against xss attacks, however some xss attacks might work by sending it through the messages however only sent by the user. It does not pop up again as it is not saved correctly in the database.

Before this function was introduced we had to sanitise each input, if they were not sanitised due to forgetting to code it then there was a chance of SQL injection

With the sanitisation the change is lowered.

Parametrised queries are also used to ensure that SQL injections also do not work properly.

|  |  |
| --- | --- |
| Attack Type | Protection (RAG) |
| Privledge Escalatiion |  |
| SQL injection |  |
| XSS attacks |  |

**Appendix 1: Testing Evidence**

|  |  |
| --- | --- |
| Ref: A1 |  |
| Ref A2 |  |
| Ref A3 |  |
| Ref A4 |  |
| Ref A5 |  |
| Ref A6 |  |