**Testing Plans**

**Unit Testing**

In order to check individual areas of our program we have segmented our code into blocks and will perform various unit tests that correspond to programs built with the JavaScript language.

We will be testing login validation, user interface functionality, and gameplay functionality. For login validation test cases include empty user id/password, strange characters, non existent accounts, logging in the same user twice. we expect that valid entries should allow users to login and invalid entries should be rejected.

As for user interface functionality, this logic must be isolated from all other logic. Test cases include Rendering of components and enabling/disabling of components based on user actions. Examples of this are loading of the card decks as well as drag and dropping the cards is their designated areas. We expect all objects to be corrected rendered and have appropriate responses to actions.

For gameplay functionality player turns will be tested with an emphasis on attacks. For example, we expect that Hit Points of opponents should be reduced when they are attacked. Unit tests will be implemented manually until adoption of a JavaScript specific library such as Jest.

**Regression Testing**

To minimize as many set backs as possible, tests will be conducted throughout the development process when modifications to code are made, new features are added and performance fixes occur. The approach to be taken will be prioritization of test cases as retesting all components is not feasible in the given timespan. This will be based on frequency of use for the program functionalities. Therefor emphasis will be taken on login and gameplay functionality. All bugs found are to be removed and will be expected to not reappear. Test will be completed manually in the immediate future we aim to automate this process as soon as possible with opensource tools such as Selenium which is highly recommended.

**Integration Testing**

Our goal with integration tests is to assure multiple classes work with each other. As the program is relatively small, we will take a “Big Bang Approach”. That is to say, we will test all modules at once and verify if the system functions overall.

**System Level Testing**

Tests will be made to imitate user interactions with our program. User experience will be an important metric in these tests. Game functionality should match the game instructions and descriptions and in result what the user expect when they are playing.