**Pet Seizure Tracker for iOS and Android: Testing Report**

James Eckler

Joshua Peterson

**Unit Testing**

*What*:

React Native has a built-in unit testing framework called Jest: <https://jestjs.io/docs/en/tutorial-react-native> Jest allows the creation of mock objects for testing and snapshots to compare how a certain component renders.

*Who*:

The team

*When*:

Tests should be written in the `\_\_test\_\_` directory as soon as a component is created and all tests run using the `npm test` command, or a specific test using `npm test [testname]`

*Result*:

Unfortunately we could not write unit tests for components that require access to Firestore; even simple actions such as signing in were unresponsive with Jest. A basic test that should ensure that the app renders properly displayed an error insinuating that this was an error with React itself. For any component we wrote that didn’t require database queries such as buttons or text boxes, we wrote tests that ensured they rendered properly and matched with previous snapshots.

**System Testing**

*What*:

Expo is a set of libraries that allows builds to be easily created and run on emulators or directly on physical devices through the Expo app: [https://expo.io](https://expo.io/)

*Who*:

The team

*When*:

While writing the program - updates can be seen as changes are made

*Result*:

Expo gave us an easy way to ensure that the application would work on multiple devices for both iOS and Android. One drawback is that upon reloading the application after any change is made, you are placed back at the login screen and need to log back in. As well as being slightly tedious, logging in to the system uses reads from our Firestore quota. This has not proven to be a problem for the most part, barring a bug that was fixed that would cause many reads as long as a certain element was in view.

**Usability Testing**

*What*:

Using Expo, open the application on a user’s phone and give them an instruction for what they should do (i.e. register an account, create a log, add a pet).

*Who*:

Someone not on the team (Preferably isn’t too used to using the application)

*When*:

After completion of major features

*Result*:

User feedback helped us refine features in a way that made the app more usable.

**Accessibility Testing**

React Native has a page detailing how to write components that are accessible to people with disabilities: <https://facebook.github.io/react-native/docs/accessibility> This allows for users to use features on their phone such as VoiceOver or TalkBack for elements to be read out loud. Features such as this allow our application to be more usable by people with visual impairments.

**Acceptance Testing**

*What*:

Upon completion, we gave the application to our client to ensure we met the requirements laid out for us.

*Who*:

Lindsey Peterson

*When*:

Project completion

*Result*:

The client went through the finished application and determined that we fulfilled the requirements laid out for us, barring a couple of features that were pushed back until next release.