Working Prototype Known Problems Report
Event Hub
3/8/2022

Known Problems:

- Console Warning that the AsyncStorage package will not be supported in the future.
  - **Action:** Warning appears upon opening the app.
  - **Location of the fault:** Unknown, possibly caused by an import of the AsyncStorage package within an Expo module used by our app
  - **Solution:** Find the faulty import statement and update it to use the new module
- The bottom of the Attending screen is cut off.
  - **Action:** The bottom of the feed on the Attending page is covered by the tab bar upon navigating to the Attending page. If one clicks on the Hosting tab and then clicks back on the Attending tab, this failure disappears.
  - **Location of the fault:** Unknown, the styling is exactly the same for the Hosting and Attending page yet the failure only occurs in the Attending page
  - **Solution:** Experiment with the style and see if we can make the failure go away.
- Users are able to create an event that starts and ends at the same time.
  - **Action:** After going to the Create screen, you are able to create an event with the same start and end time.
  - **Location of the fault:** Incomplete logic and error checking in the Create screen.
  - **Solution:** Automatically set the end date to be an hour after the start date when the start date is changed; add an error message to the event validation if the user tries to make the end time the same as the start time
- On the Event Create Screen, clicking outside the keyboard when editing fields doesn't cause it to disappear.
  - **Action:** After going to the Create screen and clicking on a text input component, clicking outside of the component has no effect on the keyboard.
  - **Location of the fault:** Incomplete event handling on the Create screen
  - **Solution:** Wrap the Create screen in a Touchable component which dismisses the keyboard when clicked.
- Image loading speed is sometimes slow throughout the app.
  - **Action:** When navigating to one of the event feeds or the Profile screen, the images sometimes have long and inconsistent loading times.
  - **Location of the fault:** Unknown, possibly because of excessive re-rendering on the various screens which may be causing the app to request the images multiple times.
  - **Solution:** Locate and fix the cause of the excessive re-rendering on the different screens. If this has no effect on the failure, look into the fast-image-react-native package.
- “Unable to perform state updates on an unmounted component” error on Feed Screen.
  - **Action:** On the Event Feed screen, clicking on the search bar and then clicking outside of it to dismiss the keyboard causes this warning to appear.

    - **Location of the fault:** Unknown, the unmounted component is likely the location search bar that we are hiding when the keyboard is dismissed.
    - **Solution:** Research this error message; try to follow the instructions that came with the error message: “To fix, cancel all subscriptions and asynchronous tasks in a useEffect cleanup function.”
- Unable to perform state updates on an unmounted component” error after editing an event.
    - **Action:** After editing an event and clicking save, this warning appears while navigating to the Event Details screen.
    - **Location of the fault:** Unknown, the unmounted component is likely the Edit Event screen which is being navigated away from and it possibly occurs when we reset the input fields on this screen.
    - **Solution:** Research this error message; try to follow the instructions that came with the error message: “To fix, cancel all subscriptions and asynchronous tasks in a useEffect cleanup function.”
- In Create Screen, must use a timeout so prevent the Expo app from crashing upon creating events
    - **Action:** The timeout is hiding this error but if it is removed and you select an image to upload, clicking the “Create Event” button will cause the Expo app to crash.
    - **Location of the fault:** The fault is in the uploadImageAsync() function on the Create Screen. If there is no timeout between the line which creates a reference to the image in firestore and the line which uploads the image to firestore, the Expo app will crash.
    - **Solution:** Try to run the app in a non-Expo environment without the timeout and see if the failure persists. If it persists, try to refactor the uploadImageAsync() to prevent the crash.
- In Edit Profile Screen, must use a timeout so prevent the Expo app from crashing upon creating events
    - **Action:** The timeout is hiding this error but if it is removed and you select an image to upload, clicking the “Save” button will cause the Expo app to crash.
    - **Location of the fault:** The fault is in the uploadImageAsync() function on the Edit Profile Screen. If there is no timeout between the line which creates a reference to the image in firestore and the line which uploads the image to firestore, the Expo app will crash.
    - **Solution:** Try to run the app in a non-Expo environment without the timeout and see if the failure persists. If it persists, try to refactor the uploadImageAsync() to prevent the crash.
- “VirtualizedList: You have a large list that is slow to update” warning on event feeds
    - **Action:** The warning occasionally appears on the main Event Feed screen, the Attending screen, and the Hosting Screen.
    - **Location of the fault:** Possibly because of the complexity of the EventCard renderItem Component we are using in these FlatLists.

    - **Solution:** Research the error, try to convert the renderItem into a PureComponent as per the provided instructions: “make sure your renderItem function renders components that follow React performance best practices like PureComponent, shouldComponentUpdate, etc”
- Nested VirtualList warning in the Create screen and Edit Event Screen
    - **Action:** Click on the location input in the Create screen or Edit Event Screen.
    - **Location of the fault:** The GooglePlacesAutocomplete component we are using for the location input on these screens uses a VirtualList, which should not be nested within a ScrollView component.
    - **Solution:** Determine if there is a prop we can pass to ScrollView or another similar component we can use instead that will cause the message to go away.