Name:Hayden Cordeiro
BECOMPS
RollNo:05
Batch:D

Experiment 14

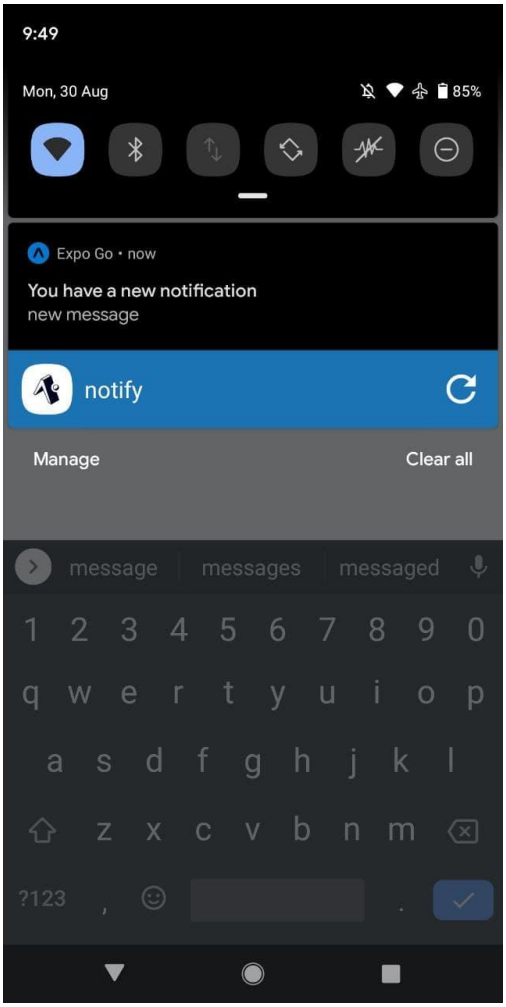
Aim:
To Implement an Android Application that creates an alert upon receiving a message.
Theory:
A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer. Both the notification area and the notification drawer are system- controlled areas that the user can view at any time. Android Toast class provides a handy way to show users alerts but problem is that these alerts are not persistent which means alert flashes on the screen for a few seconds and then disappears.
Code:

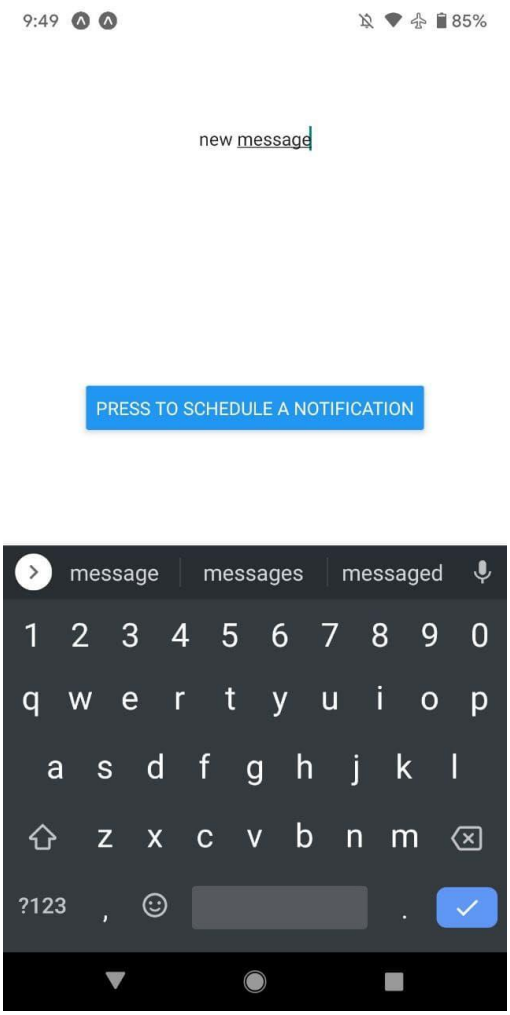
```
import Constants from 'expo-constants';
import * as Notifications from 'expo-notifications';
import React, { useState, useEffect, useRef } from 'react';
import { Text, View, Button, Platform,TextInput } from 'react-native';

Notifications.setNotificationHandler({
  handleNotification: async () => ({
    shouldShowAlert: true,
    shouldPlaySound: false,
    shouldSetBadge: false,
  }),
});

export default function App() {
  const [expoPushToken, setExpoPushToken] = useState('');
  const [notification, setNotification] = useState(false);
  const [title, setTitle] = useState("");
  const notificationListener = useRef();
  const responseListener = useRef();

  useEffect(() => {
    registerForPushNotificationsAsync().then(token => setExpoPushToken(token));

    notificationListener.current = Notifications.addNotificationReceivedListener(
notification => {
      setNotification(notification);
    });
```

```
    responseListener.current = Notifications.addNotificationResponseReceivedListe
ner(response => {
      console.log(response);
    });

    return () => {
      Notifications.removeNotificationSubscription(notificationListener.current);
      Notifications.removeNotificationSubscription(responseListener.current);
    };
  }, []);

  return (
    <View
      style={{
        flex: 1,
        alignItems: 'center',
        justifyContent: 'space-around',
      }}>

      <View style={{ alignItems: 'center', justifyContent: 'center' }}>

      <TextInput
               underlineColorAndroid = "transparent"
               placeholder = "Text here"
               placeholderTextColor = "#9a73ef"
               autoCapitalize = "none"
               onChangeText = {(text)=>setTitle(text)}/>
      </View>
      <Button
        title="Press to schedule a notification"
        onPress={async () => {
          await schedulePushNotification(title);
        }}
      />
    </View>
  );
}

async function schedulePushNotification(title) {
  await Notifications.scheduleNotificationAsync({
    content: {
      title: "You have a new notification",
      body: title,
      // data: { data: 'goes here' },
    },
```

```
    trigger: { seconds: 1 },
  });
}

async function registerForPushNotificationsAsync() {
  let token;
  if (Constants.isDevice) {
    const { status: existingStatus } = await Notifications.getPermissionsAsync();
    let finalStatus = existingStatus;
    if (existingStatus !== 'granted') {
      const { status } = await Notifications.requestPermissionsAsync();
      finalStatus = status;
    }
    if (finalStatus !== 'granted') {
      alert('Failed to get push token for push notification!');
      return;
    }
    token = (await Notifications.getExpoPushTokenAsync()).data;
    console.log(token);
  } else {
    alert('Must use physical device for Push Notifications');
  }

  if (Platform.OS === 'android') {
    Notifications.setNotificationChannelAsync('default', {
      name: 'default',
      importance: Notifications.AndroidImportance.MAX,
      vibrationPattern: [0, 250, 250, 250],
      lightColor: '#FF231F7C',
    });
  }

  return token;
}
```

Output:





Conclusion:

In this experiment I learnt to work with React Native I learnt about notifications and why they might be useful in an app. Also implemented an app that creates notifications and we tested it on an Android phone and the expected output was achieved.