

```

import React, { Component } from 'react';
import {
  StyleSheet,
  View,
  Text,
  Alert
} from 'react-native';
import firebase, { database } from 'firebase';
import { Icon } from 'react-native-elements';
import Geocoder from 'react-native-geocoding';
import { FlatList, ScrollView } from 'react-native-gesture-handler';

import AsyncImage from '../components/AsyncImage';
import ButtonComponent from '../components/ButtonComponent';

class DisplayClaim extends Component {

  constructor(props) {
    super(props)

    this.state = {
      id: this.props.navigation.getParam("id"),
      claim_info: null,
      loading: true
    }
  }

  async componentDidMount(){
    console.log(this.state)
    try {
      //EVENT: firebase call
      //fetch claim information /users/<uid>/claims/claim_id
      const ref = firebase.database().ref(`/users
/${firebase.auth().currentUser.uid}/claims/${this.state.id}/`)
      const claim_info = await ref.once('value')
        .then(snapshot => {
          return snapshot.val()
        })
        .catch( error => Toast.show(error))

      this.setState({claim_info: claim_info, loading: false})
    } catch (error) {
      console.warn("Error fetching data ----- ", error)
    }
  }

  //delete claim
  deleteClaim(){
    const { claim_info, id } = this.state

    //updat firebase db
    firebase.database().ref(`/users/${firebase.auth().currentUser.uid}/claims/${id}
/`).remove()
      .catch(e => console.warn(e))

    //update cloud storage
    //remove image
    firebase.storage().ref(`/uploads/${firebase.auth().currentUser.uid}
/${claim_info.quote_image_id}.jpeg`).delete()
      .catch(e => console.warn(e))
  }
}

```

```

        //update cloud storage
        //remove image
        firebase.storage().ref(`/uploads/${firebase.auth().currentUser.uid}
/${claim_info.report_image_id}.jpeg`).delete()
            .catch(e => console.warn(e))

        //navigate back to view claims
        this.props.navigation.navigate('ViewClaims')
    }

    render() {
        const { loading, claim_info } = this.state
        console.log(claim_info)
        if(!loading){
            return(
                <View style={styles.container}>
                    <Text style={styles.logo}>Claim Details</Text>
                    <Text>Submitted on {claim_info.date_submitted}</Text>
                    { claim_info.complete ? (
                        <View>
                            <Text>This claim has been {claim_info.outcome}</Text>
                            <Text>Reviewed on {claim_info.date_reviewed}</Text>
                        </View>
                    ) : (
                        <Text>Pending</Text>
                    )}
                    <Text>Claim for the amount of €{claim_info.amount}</Text>
                    <ScrollView>
                        <Text style={styles.contentHeader}>Contacts</Text>
                        <FlatList
                            data={claim_info.contact_list}
                            renderItem={({item, index}) =>
                                <View>
                                    <View style={{paddingLeft: '10%',
borderBottomWidth: 1, borderBottomColor: '#2E6CB5', paddingBottom: 10}}>
                                        <Text style={{fontSize: 18, fontWeight:
'bold'}}>{item.role}</Text>
                                        <Text style={{fontSize: 18}}>Name:
{item.name}</Text>
                                        <Text style={{fontSize: 18}}>Phone Number:
{item.phone}</Text>
                                        <Text style={{fontSize: 18}}>Email:
{item.email}</Text>
                                    </View>
                                </View>
                            }
                            keyExtractor={(item, index) => index.toString()}
                        />
                        <Text style={styles.contentHeader}>Description</Text>
                        <Text style={styles.content}>{claim_info.description}</Text>
                        <Text style={styles.contentHeader}>Quote</Text>
                        <AsyncImage
                            id={claim_info.quote_image_id}
                            style={{width: '100%', height: 400, resizeMode:
'contain'}}

```

```

        refresh={this.state.refresh}
        uid={firebase.auth().currentUser.uid}
      />

      <Text style={styles.contentHeader}>Report</Text>
      <AsyncImage
        id={claim_info.report_image_id}
        style={{width: '100%', height: 400, resizeMode:

'contain'}}

        refresh={this.state.refresh}
        uid={firebase.auth().currentUser.uid}
      />
    </ScrollView>

    { !claim_info.complete ? (
      <ButtonComponent
        icon='delete'
        type='antdesign'
        onPress={() => {
          Alert.alert(
            'Delete Claim',
            'Are you sure you would like to delete claim?',
            [
              {
                text: 'Cancel',
                style: 'cancel',
              },
              {
                text: 'Delete',
                onPress: () => this.deleteClaim(),
              },
            ],
          );
        }}
        text='Delete'
      />
    ) : (null)}

    <ButtonComponent
      icon='arrowleft'
      type='antdesign'
      onPress={() => this.props.navigation.goBack()}
      text='Back'
    />
  </View>
)
} else {
  return(
    <View style={styles.container}>
      <View>
        <Text style={styles.logo}>Loading Profile...</Text>
      </View>
    </View>
  )
}
}

```

```
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    width: '100%'
  },
  logo:{
    fontWeight:"bold",
    fontSize:35,
    color:"#2E6CB5",
    marginBottom:10,
    paddingTop: '10%'
  },
  contentHeader: {
    color: '#2E6CB5',
    fontSize: 28,
    fontWeight: 'bold',
    paddingTop: 20,
    paddingLeft: '5%',
    paddingBottom: 10
  },
  content: {
    fontSize: 20,
    paddingHorizontal: '5%'
  },
})

export default DisplayClaim
```