```
import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View,
  ActivityIndicator
} from 'react-native';
import { ScrollView, FlatList, TouchableOpacity } from 'react-native-gesture-handler';
import firebase from 'firebase'
import Toast from 'react-native-simple-toast';
import { Pages, Indicator } from 'react-native-pages';
import Claim from '../../components/Claim'
import humanize from 'humanize-plus';

class ClaimsScreen extends Component {
  constructor(props) {
    super(props);
    this.state = {
        pending_claims: [],
        complete_claims: [],
        loading: true
    }
  }

  async componentDidMount(){
    try {
        //EVENT: firebase call
        //fetch claims list
        await firebase.database().ref(`/users/`).once('value')
            .then(snapshot => {
                var pending_list = []
                var complete_list = []
                snapshot.forEach((element) => {
                    if(element.val().claims !== undefined) {
                        const uid = element.key
                        const user_claims = element.val().claims
                        Object.keys(user_claims).forEach(function(key) {
                            //seperate pending and complete claims
                            if(user_claims[key].complete){
                                complete_list.push({
                                    uid: uid,
                                    key: key,
                                    claim: user_claims[key]
                                })
                            } else {
                                pending_list.push({
                                    uid: uid,
                                    key: key,
                                    claim: user_claims[key]
                                })
                            }
                          });
                    }
                })
                this.setState({ pending_claims: pending_list, complete_claims:
complete_list })
            }).catch(error => console.warn(error))

            this.setState({ loading: false})
```

```
    } catch (error) {
        console.warn("Error fetching data ---------------------------- ", error)
    }
   }

  render() {
    const { loading, pending_claims, complete_claims } = this.state
    return (
        <Pages>
            <View style={styles.container}>
                <View style={styles.headerContainer}>
                    <View>
                        <Text style={styles.logo}>Pending Claims</Text>
                    </View>
                </View>
                <View style={styles.contentContainer}>
                    { !loading ? (
                        <View>
                            <Text style={{alignSelf: 'flex-end', paddingRight:
10}}>{this.state.pending_claims.length}
{humanize.pluralize(this.state.pending_claims.length, "pending claim")}</Text>
                            <ScrollView>
                                <FlatList
                                    data={pending_claims}
                                    renderItem={({item, index}) =>
                                        <TouchableOpacity onPress={() => {

this.props.navigation.navigate("ReviewClaim", item)
                                        }}>
                                            <Claim
                                                date={item.claim.date_submitted}
                                                id={item.key}
                                                status={null}
                                                amount={null}
                                            />
                                        </TouchableOpacity>
                                        }
                                    keyExtractor={(item, index) => index.toString()}
                                />
                            </ScrollView>
                        </View>
                    ) : (
                        <View style={{ flex: 1, justifyContent: 'center', alignItems:
'center' }}>
                            <ActivityIndicator
                                size='large'
                                color='#2E6CB5'
                            />
                            <Text style={{ fontSize: 20, paddingTop: 20}}>Loading Data
...</Text>
                        </View>
                    )}

                </View>
            </View>
            <View style={styles.container}>
                <View style={styles.headerContainer}>
                    <View>
                        <Text style={styles.logo}>Complete Claims</Text>
```

```
                    </View>
                </View>

                <View style={styles.contentContainer}>
                    { !loading ? (
                        <View>
                            <Text style={{alignSelf: 'flex-end', paddingRight:
10}}>{this.state.complete_claims.length}
{humanize.pluralize(this.state.complete_claims.length, "completed claim")}</Text>
                            <ScrollView>
                                <FlatList
                                    data={complete_claims}
                                    renderItem={({item, index}) =>
                                        <Claim
                                            date={item.claim.date_submitted}
                                            id={item.key}
                                            status={true}
                                            outcome={item.claim.outcome}
                                            amount={item.amount}
                                        />
                                    }
                                    keyExtractor={(item, index) => index.toString()}
                                />
                            </ScrollView>
                        </View>
                    ) : (
                        <View style={{ flex: 1, justifyContent: 'center', alignItems:
'center' }}>
                            <ActivityIndicator
                                size='large'
                                color='#2E6CB5'
                            />
                            <Text style={{ fontSize: 20, paddingTop: 20}}>Loading Data
...</Text>
                        </View>
                    )}

                </View>
            </View>
        </Pages>
    )}
}

const styles = StyleSheet.create({
  container: {
      flex: 1
  },
  headerContainer: {
    backgroundColor: '#EFF1F3',
    flex: 3,
    flexDirection: 'row',
    alignItems: 'flex-end',
    padding: 20
  },
  contentContainer:{
    flex: 6,
    padding: 20,
    justifyContent: 'space-between',
    backgroundColor: 'white'
```

```
  },
  text: {
    fontSize: 16
  },
  logo:{
    fontWeight:"bold",
    fontSize:35,
    color:"#2E6CB5",
    marginBottom:40
  }
});

export default ClaimsScreen;
```