```
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 Claim from '../../components/Claim'

class ViewClaims extends Component {
  constructor(props) {
    super(props);
    this.state = {
        claims: [],
        pending_count: 0
    }
  }

  async componentDidMount(){
    try {
        //EVENT: firebase call
        //fetch claims list
        const claims_list = await firebase.database().ref(`/users
/${firebase.auth().currentUser.uid}/claims/`).once('value')
            .then(snapshot => {
                var temp_list = []
                var count = 0
                snapshot.forEach((element) => {
                    const data = {
                        id: element.key,
                        date: element.val().date_submitted,
                        complete: element.val().complete,
                        outcome: element.val().outcome,
                        description: element.val().description,
                        quote_image_id: element.val().quote_image_id,
                        report_image_id: element.val().report_image_id,
                        amount: element.val().amount
                    }
                    temp_list.push(data)

                    if(!element.val().complete){
                        count++
                    }
                })
                this.setState({pending_count: count})

                return temp_list

            }).catch(error => Toast.show(error))

            this.setState({ claims: claims_list.reverse(), loading: false})
    } catch (error) {
        console.warn("Error fetching data ---------------------------- ", error)
    }
   }
  render() {
    const { loading, claims } = this.state
    return (
        <View style={styles.container}>
            <View style={styles.headerContainer}>
                <View>
                    <Text style={styles.logo}>My Claims</Text>
                    <Text style={styles.text}>You can view the status of your claims
below.</Text>
                </View>
            </View>
            <View style={styles.contentContainer}>
                { !loading ? (
                    <View>
                        <Text style={{alignSelf: 'flex-end', paddingRight:
10}}>{this.state.pending_count}/{this.state.claims.length} pending claims</Text>
                        <ScrollView>
                            <FlatList
                                data={claims}
                                renderItem={({item, index}) =>
                                    <View>
                                        <TouchableOpacity onPress={() => {

this.props.navigation.navigate("DisplayClaim", item)
                                        }}>
                                            { item.complete ? (
                                                <Claim
                                                    date={item.date}
                                                    amount={item.amount}
                                                    id={item.id}
                                                    status={item.complete}
                                                    outcome={item.outcome}
                                                />
                                            ) : (
                                                <Claim
                                                    date={item.date}
                                                    amount={item.amount}
                                                    id={item.id}
                                                    status={item.complete}
                                                />
                                            )}
                                        </TouchableOpacity>

                                    </View>
                                }
                                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>
    )}
}

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,
    paddingHorizontal: 20
  },
  logo:{
    fontWeight:"bold",
    fontSize:35,
    color:"#2E6CB5",
    marginBottom:40,
    paddingHorizontal: 20
  }
});

export default ViewClaims;
```