

```

//TODO analytics lines chart

import React, { Component } from 'react';
import {
  StyleSheet,
  View,
  Text,
  Dimensions,
  RefreshControl,
  ActivityIndicator
} from 'react-native';
import firebase from 'firebase';
import humanize from 'humanize-plus';
import Toast from 'react-native-simple-toast';

import Journey from '../components/Journey'
import { ScrollView, FlatList } from 'react-native-gesture-handler';

const screenWidth = Dimensions.get("window").width;

class HomeScreen extends Component {
  constructor(props) {
    super(props);
    this.state = {
      totalAmount: 0,
      showJourneys: true,
      list: [],
      user_info: {},
      loading: true,
      refreshing: false
    };
  }

  async componentDidMount(){
    const { currentUser } = firebase.auth()
    try {
      //fetch user information /users/<uid>
      const ref = firebase.database().ref(`/users/${currentUser.uid}`)
      const user_info = await ref.once('value')
        .then(snapshot => {
          return snapshot.val()
        })
        .catch( error => Toast.show(error))

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

    try {
      var amount = 0
      //fetch past journeys /users/<uid>/journeys
      const db_list = await firebase.database().ref(`/users/${currentUser.uid}
/journeys/`).once('value')
        .then(snapshot => {
          var journey_list = []
          var currentMonth = this.humanizedMonth(new Date().getMonth())
          snapshot.forEach((childSub) => {
            //calculate cost for this month
            if(currentMonth == childSub.val().billing_month){

```

```

        amount = amount + childSub.val().cost
      }
      //push journey to list
      journey_list.push(childSub.val())
    })

    return journey_list
  }).catch(error => {
    Toast.show(error)
  })

  this.setState({totalAmount: amount, list: db_list.reverse(), loading:
false})
} catch (error) {
  console.warn("Error fetching data ----- ", error)
}

}

//event listener
//on user refresh list
async refreshList(){
  const { currentUser } = firebase.auth()
  try {
    var amount = 0
    //fetch db journeys list
    const db_list = await firebase.database().ref(`/users/${currentUser.uid}
/journeys/`).once('value')
      .then(snapshot => {
        var journey_list = []
        var currentMonth = this.humanizedMonth(new Date().getMonth())
        snapshot.forEach((childSub) => {
          if(currentMonth == childSub.val().billing_month){
            amount = amount + childSub.val().cost
          }
          journey_list.push(childSub.val())
        })

        return journey_list
      }).catch(error => {
        Toast.show(error)
      })

    this.setState({totalAmount: amount, list: db_list.reverse(), refreshing:
false})
  } catch (error) {
    console.warn("Error fetching data ----- ", error)
  }
}

//build Journey component
renderJourney(distance, cost){
  return <Journey distance={distance} cost={cost} />
}

//return human friendly month string
humanizedMonth(month){
  switch (month) {
    case 0:

```

```

        month = "Jan"
        break;
    case 1:
        month = "Feb"
        break;
    case 2:
        month = "Mar"
        break;
    case 3:
        month = "Apr"
        break;
    case 4:
        month = "May"
        break;
    case 5:
        month = "Jun"
        break;
    case 6:
        month = "Jul"
        break;
    case 7:
        month = "Aug"
        break;
    case 8:
        month = "Sep"
        break;
    case 9:
        month = "Oct"
        break;
    case 10:
        month = "Nov"
        break;
    case 11:
        month = "Dec"
        break;
    default:
        break;
    }
    return month
}

```

```

render() {
    const {loading, refreshing, list} = this.state
    return(
        <View style={styles.main}>
            <View style={styles.headerContainer}>
                <View style={{flex:1}}>

                    </View>
                    <View>
                        <Text style={styles.welcome}>Welcome
{this.state.user_info.firstname}</Text>
                    </View>

                <View style={styles.amount}>
                    <View>
                        <Text style=
{styles.amountHeader}>€{this.state.totalAmount.toFixed(2)}</Text>

```

```

        </View>

        <View>
            <Text style={styles.subheading}> Monthly bill</Text>
        </View>
    </View>
    <View style={styles.journeysContainer}>
        <Text style={styles.journeysHeader}>Past Journeys</Text>
        {
            !loading ? (
                <View>
                    <Text style={{alignSelf: 'flex-end', paddingRight:
20}}>{this.state.list.length} {humanize.pluralize(this.state.list.length, "past
journey")}</Text>

                    <ScrollView
                        refreshControl={
                            <RefreshControl refreshing={refreshing}
onRefresh={() => {
                                this.setState({refreshing: true})
                                this.refreshList()}
                            } />
                        >
                            <FlatList
                                data={this.state.list}
                                renderItem={({item, index}) =>
                                    <Journey
                                        style={styles.item}
                                        address={item.address}
                                        date={item.humanized_date}
                                        distance={item.distance}
                                        cost={item.cost}
                                        nightdrive={item.nightdrive}
                                        vehiclename={item.vehiclename}
                                    />
                                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>
)
}
}

```

```

//lower header

const styles = StyleSheet.create({
  main: {
    flex: 5,
    flexDirection: 'column'
  },
  headerContainer:{
    flex: 2,
    paddingTop: '12%',
    alignItems: 'center',
    justifyContent: 'space-around'
  },
  subheading: {
    fontSize: 15,
    paddingBottom: 10,
  },
  welcome:{
    fontWeight:"bold",
    fontSize:38,
    color:"#2E6CB5",
  },
  amount:{
    flex: 2,
    flexDirection: 'row',
    alignItems: 'flex-end',
    alignSelf: 'center',
  },
  analytics:{
    flex: 2,
  },
  amountHeader:{
    paddingTop: 20,
    textAlign: 'center',
    fontSize: 45,
    textAlignVertical: 'bottom',
    color: '#EFC066'
  },
  journeysContainer:{
    flex: 5,
    backgroundColor: 'white'
  },
  journeysHeader: {
    textAlign: 'center',
    fontSize: 24,
    paddingTop: 15,
    fontWeight: 'bold',
    color: '#2E6CB5'
  }
});

export default HomeScreen;

```