

Grupa	PS3
Imię I nazwisko studenta	Artur Tagisow
Numer I tytuł zadania	2 – Aplikacje bazodanowe w React Native

1. Teoria

Do implementacji bazy danych sqlite użyto paczki NPM ‘expo-sqlite’. Paczka ta została napisana dawno temu, co można zauważyć po rozwiązaniach technologicznych. Paczka nie używa Promise, a wykorzystuje tradycyjne callbacki (tj. `ExecuteSql()` => `{console.log(‘success’), () => {console.warn(‘failure’)}}`).

Jako ćwiczenie, stworzono własny plik z serwisem, który przekształca podejście z użyciem “callback” na “Promise” (fragment):

```

public getCalculations() {
  return new Promise < GetCalculationResult[] > ((resolve, reject) => {
    this.executeSql(
      SELECT rowid, * FROM temp.calculations;
    , [], (t, results) => {
      console.log(results.rows);
      resolve(( < any > results.rows)._array as unknown as
GetCalculationResult[])
    }, (t, e) => {
      reject();
      return false
    }
  )
})

```

Sqlite nie odbiega składnią znacząco od znanych nam systemów SQL, dlatego też rozpoczęcie pracy z SQLite było łatwe.

Firebase (właściwie Firebase Remote Database) jest bazą zdalną stworzoną przez Google. Korzystanie z niej wymaga stworzenia bazy danych w serwisie internetowym. Użyto paczki NPM ‘firebase’

Do stworzenia widoków aplikacji wykorzystano standardowe komponenty np:

- TouchableOpacity
- Button
- Picker
- TextInput

Do nawigacji standardowo użyto paczki NPM ‘react-navigation’

Listingi

Aplikacja zawiera 3 widoki:

- **widok dodawania samochodów**
użyto elementów Picker

←

Add

Ford

/QjOjEyMDd9&auto=format&fit=crop&w=1000&q=80

Benzyna

3

100

DODAJ

18:52

80

←

Add

Ford

Benzyna

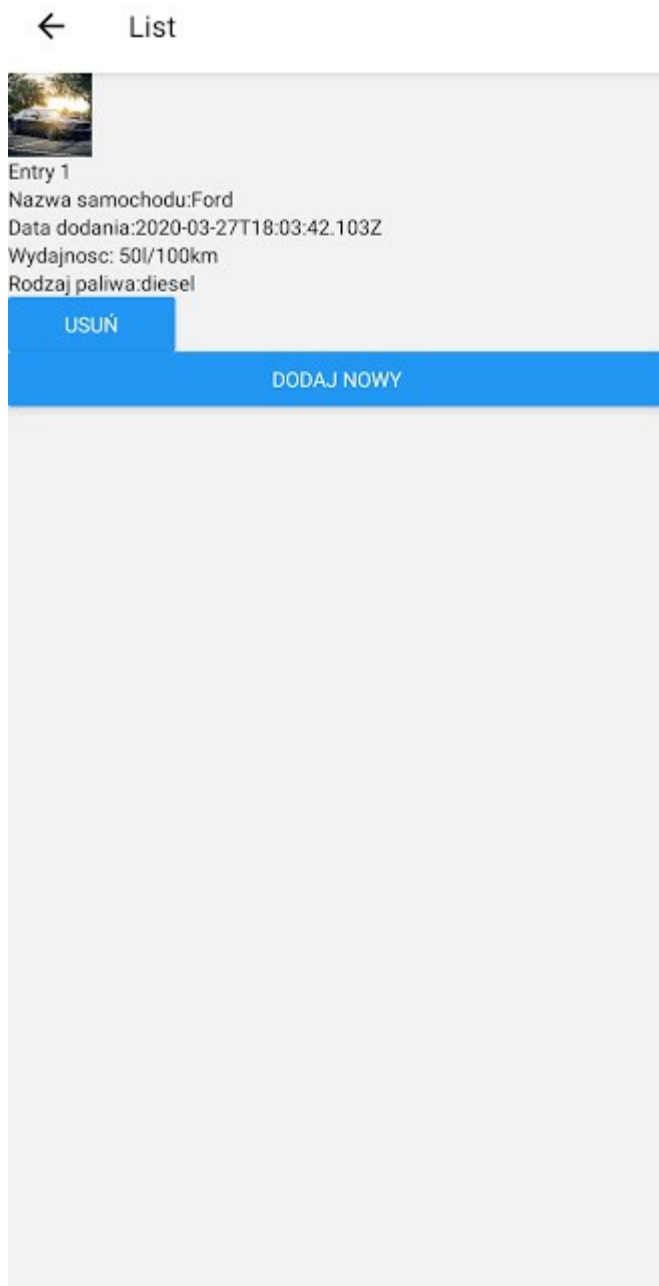
Diesel

kWhs

100

DODAJ

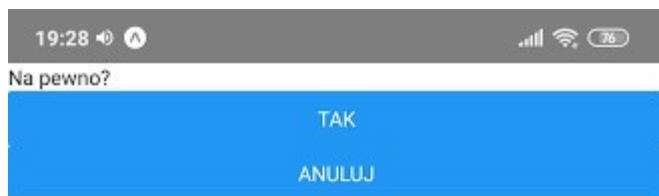
- widok wyświetlający dodane samochody
Elementy <Text>, <Button>, <View> owrapowane <TouchableOpacity>



Kod elementu listy aut

```
{...}  
  <TouchableOpacity onPress={() => { this.props.navigation.push("Details", {car_name,  
creation_date, fuel_efficiency, fuel_type, image_url, rowid}) }} key={rowid}>  
    <View>  
      <Image source={{ uri: image_url }} style={style.tinyLogo}></Image>  
      <Text>Entry {rowid}</Text>  
      <Text>Nazwa samochodu:{car_name}</Text>  
      <Text>Data dodania:{creation_date}</Text>  
      <Text>Wydajnosć: {fuel_efficiency}l/100km</Text>  
      <Text>Rodzaj paliwa:{fuel_type}</Text>  
      <TouchableOpacity style={style.button}>  
        <Button title="Usuń" onPress={() => {  
          this.setState({modalVisible: rowid})  
        }}>  
      </TouchableOpacity>  
    </View>  
  </TouchableOpacity>  
}
```

Widok po wciśnięciu “Usuń” na którymś z samochodów
Użyto elementu <Modal> z paczki ‘react-native’



- **widok szczegółów danego samochodu**



Details



Nazwa: Ford

Utworzono: 2020-03-27T18:03:42.103Z

Wydajność: 50 l/km

Rodzaj paliwa: diesel

Ford

Ford

Implementacje Firebase I SQLite

SQLite

Inicjalizacja tabel w SQLite:

// poniższy kod znajduje się w klasie

```
private createTables() {  
    return new Promise<void>((resolve, reject) => {  
        this.executeSql(  
            CREATE TABLE IF NOT EXISTS temp.calculations (  
                car_name varchar(200),  
                image_url varchar(200),  
                fuel_type varchar(200),  
                fuel_efficiency varchar(200),  
                creation_date varchar(200)  
            )  
            , [], () => { resolve(); }, (t, e) => { reject(e); return false; });  
    })  
}
```

Dodanie do bazy danych:

```
public addCalculation(carName: string, imageUrl: string, fuelType: 'diesel' |  
'petrol', fuelEfficiency: string, creationDate: string) {  
    return new Promise<void>((resolve, reject) => {  
        this.executeSql(  
            INSERT INTO temp.calculations (  
                car_name,  
                image_url,  
                fuel_type,  
                fuel_efficiency,  
                creation_date  
            ) VALUES (  
                ?,  
                ?,  
                ?,  
                ?,  
                ?  
            )  
            , [carName, imageUrl, fuelType, fuelEfficiency, creationDate], () =>  
            { resolve() }, (t, e) => { reject(e); return false; })  
    })  
}
```

```
}
```

Pobranie danych z SQLite:

```
public getCalculations() {  
    return new Promise<GetCalculationResult[]>((resolve, reject) => {  
        this.executeSql(  
              
            SELECT rowid, * FROM temp.calculations;  
            , [], (t, results) => { console.log(results.rows);  
resolve((<any>results.rows)._array as unknown as GetCalculationResult[]) }, (t, e)  
=> { reject(); return false}  
        )  
    })  
}
```

Usunięcie z SQLite

```
public deleteCalculation(rowid: number) {  
    return new Promise<GetCalculationResult[]>((resolve, reject) => {  
        this.executeSql(  
              
            DELETE FROM temp.calculations WHERE rowid = ?;  
            , [rowid], (t, results) => { console.log(results.rows);  
resolve((<any>results.rows)._array as unknown as GetCalculationResult[]) }, (t, e)  
=> { reject(e); return false}  
        )  
    })  
}
```


Firestore

```
import * as firebase from "firebase";
```

```
const firebaseConfig = {  
  apiKey: "AIzaSyCPVmlQharW8Ne0Mm_74x3HqwsDoZ1mAYS",  
  authDomain: "mobilki-d7662.firebaseio.com",  
  databaseURL: "https://mobilki-d7662.firebaseio.com",  
  projectId: "mobilki-d7662",  
  storageBucket: "mobilki-d7662.appspot.com",  
  messagingSenderId: "341261303541",  
  appId: "1:341261303541:web:d1e979b7d57c26ddbdcfbc"  
};
```

```
class FirebaseService {  
  private firebase = firebase.initializeApp(firebaseConfig).database();  
  //dodawanie samochodu do bazy  
  setCar(car_name: string, image_url: string, fuel_type: string, fuel_efficiency:  
string, creation_date: string) {  
    return this.firebase.ref('cars').push({  
      car_name,  
      image_url,  
      fuel_type,  
      fuel_efficiency,  
      creation_date  
    })  
  }  
  //pobieranie samochodów z bazy  
  getCars() {  
    return this.firebase.ref('cars').once('value');  
  }  
  //usuwanie samochodu z bazy  
  deleteCar(id) {  
    return this.firebase.ref('cars/${id}').remove()  
  }  
}  
//export "singletona" serwisu  
export const firebaseService = new FirebaseService();
```