



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Контрольная работа № 2.2
по курсу «Разработка мобильных приложений»
«HTTP JSON parser»

Студент группы ИУ9-71Б Яровикова А. С.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель

Реализовать виджет Яндекс.Карт вывода объектов согласно варианта из таблицы ниже, в виджете должна отображаться Яндекс.Карта с расположенными на ней метками объектов, по клику на метку объекта должен открываться виджет с подробной информацией об объекте.

2 Задание

Реализовать отображение объектов на карте. Данные необходимо запросить по соответствующему URL приведенному в таблице вариантов, выполнить разбор полученных данных, представленных в формате JSON. При этом должна быть реализована возможность кликнуть на метку объекта и получить подробную информацию об объекте отмеченном на карте.

Вариант: http://pstgu.yss.su/iu9/mobiledev/lab4_yandex_map/2023.php?x=var20

3 Реализация

Исходный код представлен в листинге 1.

Листинг 1: main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:yandex_mapkit/yandex_mapkit.dart';
3 import 'package:dio/dio.dart';
4
5 void main() => runApp(MyApp());
6
7
8 class UserData {
9   final String name;
10  final String gps;
11  final String address;
12  final String tel;
13
14  UserData({required this.name, required this.gps, required this.address
    , required this.tel});
15
16  factory UserData.fromJson(Map<String, dynamic> json) {
```

```

17     return UserData(
18         name: json['name'] ?? '',
19         gps: json['gps'] ?? '',
20         address: json['address'] ?? '',
21         tel: json['tel'] ?? '',
22     );
23 }
24 }
25
26 class MapPoint {
27     final String name;
28
29
30     ///
31     final double latitude;
32
33
34     ///
35     final double longitude;
36
37     final String address;
38
39     final String tel;
40
41     const MapPoint({
42         required this.name,
43         required this.latitude,
44         required this.longitude,
45         required this.address,
46         required this.tel
47     });
48 }
49
50 class MyApp extends StatelessWidget {
51     @override
52     Widget build(BuildContext context) {
53         return const MaterialApp(
54             home: MapScreen(),
55         );
56     }
57 }
58
59 class MapScreen extends StatefulWidget {
60     const MapScreen({Key? key}) : super(key: key);
61
62     @override

```

```

63 State<MapScreen> createState() => _MapScreenState();
64 }
65
66 class _MapScreenState extends State<MapScreen> {
67     final Dio dio = Dio();
68     late YandexMapController _yandexMapController;
69
70     @override
71     void initState() {
72         super.initState();
73     }
74
75     Future<List<MapPoint>> fetchData() async {
76         try {
77             final response = await dio.get('http://pstgu.yss.su/iu9/mobiledev/
lab4_yandex_map/2023.php?x=var20');
78             if (response.statusCode == 200) {
79                 final List<dynamic> dataList = response.data;
80                 final List<UserData> userList = dataList.map((json) => UserData.
fromJson(json)).toList();
81                 print(userList);
82                 List<MapPoint> points = [];
83                 for (var place in userList) {
84                     List<String> gps = place.gps.split(',').map((s) => s.trim()).
toList();
85                     var xy = gps.map((s) => double.parse(s)).toList();
86                     var p = MapPoint(name: place.name, latitude: xy[0], longitude:
xy[1], address: place.address, tel: place.tel);
87                     points.add(p);
88                 }
89                 return points;
90             } else {
91                 throw Exception("Failed to load data");
92             }
93         } catch (e) {
94             throw Exception(e);
95         }
96     }
97
98     @override
99     void dispose() {
100         _yandexMapController.dispose();
101         super.dispose();
102     }
103
104     @override

```

```

105 Widget build(BuildContext context) {
106   return Scaffold(
107     appBar: AppBar(
108       title: const Text("yandex"),
109     ),
110     body: FutureBuilder<List<MapPoint>>(
111       future: fetchData(),
112       builder: (context, snapshot) {
113         if (snapshot.connectionState == ConnectionState.waiting) {
114           return const CircularProgressIndicator();
115         } else if (snapshot.hasError) {
116           return Text('Error: ${snapshot.error}');
117         } else {
118           return YandexMap(
119             onMapCreated: (controller) async {
120               _yandexMapController = controller;
121               await _yandexMapController.moveCamera(CameraUpdate.
122                 newCameraPosition(
123                   const CameraPosition(target: Point(latitude: 50,
124                     longitude: 20), zoom: 3)
125                 ));
126             },
127             mapObjects: getPlacemarkObjects(context, snapshot.data!),
128           );
129         }
130       },
131     );
132 }
133 List<PlacemarkMapObject> getPlacemarkObjects(BuildContext context,
134   List<MapPoint> points) {
135   List<PlacemarkMapObject> objects = [];
136   int i = 0;
137   for (var point in points) {
138     print(point.name);
139     print(point.address);
140     var newObj = PlacemarkMapObject(mapId: MapObjectId(i.toString()),
141       point: Point(latitude: point.latitude, longitude: point.
142         longitude),
143       opacity: 1,
144       onTap: (_, __) => showModalBottomSheet(context: context,
145         builder: (context) => _ModalBodyView(point: point))
146     );
147     objects.add(newObj);

```

```

146         i += 1;
147     }
148     return objects;
149 }
150 }
151
152 class _ModalBodyView extends StatelessWidget {
153     const _ModalBodyView({required this.point});
154
155     final MapPoint point;
156     @override
157     Widget build(BuildContext context) {
158         return Padding(
159             padding: const EdgeInsets.symmetric(vertical: 40),
160             child: Column(mainAxisSize: MainAxisSize.min, children: [
161                 Text(point.name, style: const TextStyle(fontSize: 20)),
162                 const SizedBox(height: 20),
163                 Text(
164                     '${point.latitude}, ${point.longitude}',
165                     style: const TextStyle(
166                         fontSize: 16,
167                         color: Colors.grey,
168                     ),
169                 ),
170                 Text(point.address, style: const TextStyle(fontSize: 20)),
171                 Text(point.tel, style: const TextStyle(fontSize: 20))
172             ]),
173         );
174     }
175 }

```

4 Результаты

Результат представлен на рисунках 1 - 5.

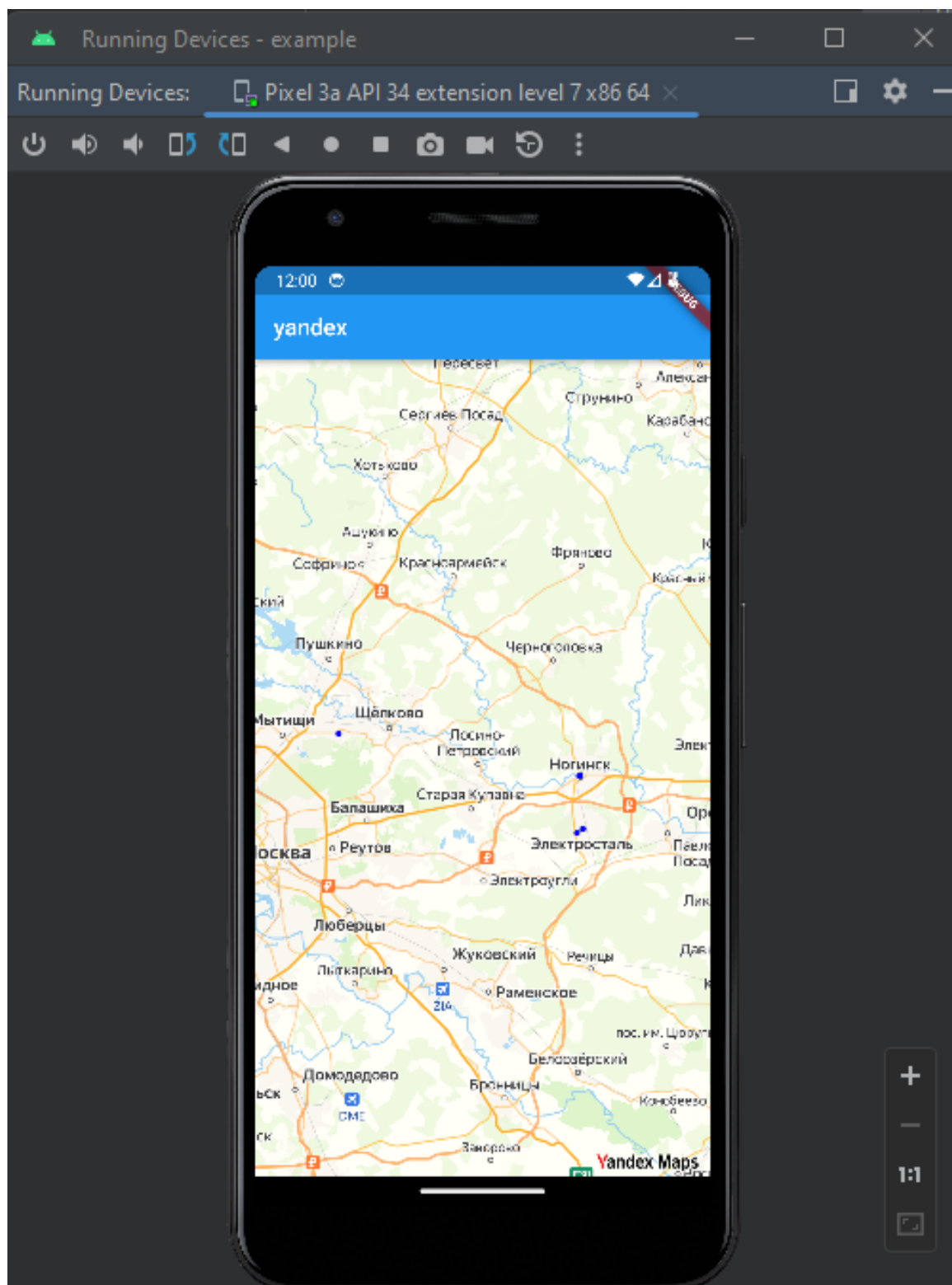


Рис. 1 — Результат работы

pstgu.yss.su/iu9/mobiledev/lab4_yandex_map/2023.php?x=var20

c","tel":"+7 (929) 663-26-63"}, {"name": "Belwer.ru", "gps": "55.856108, 38.439515", "address": "\u0420\u043e\u0433\u043e\u0436\u0441\u043a\u0430\u044f \u0443\u043b., 83, \u041d\u043e\u0433\u0438\u043d\u0441\u043a"}]

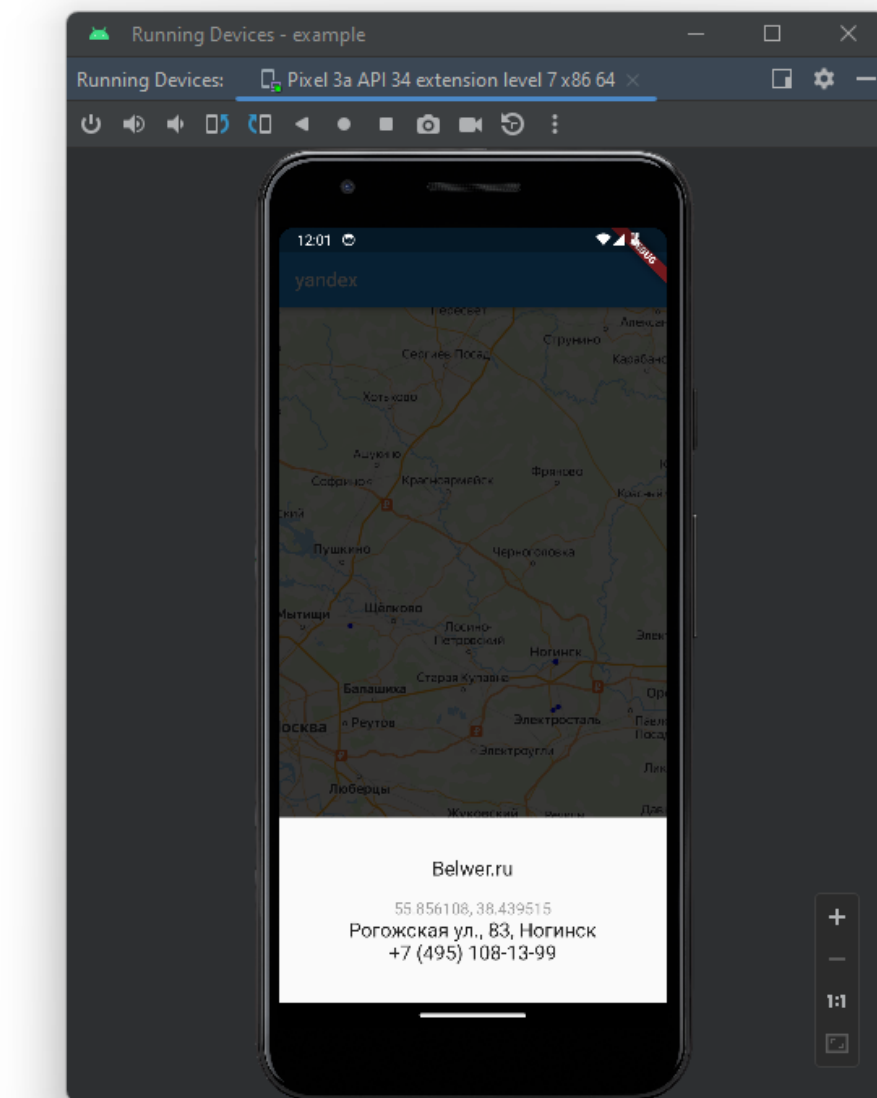


Рис. 2 — Результат работы

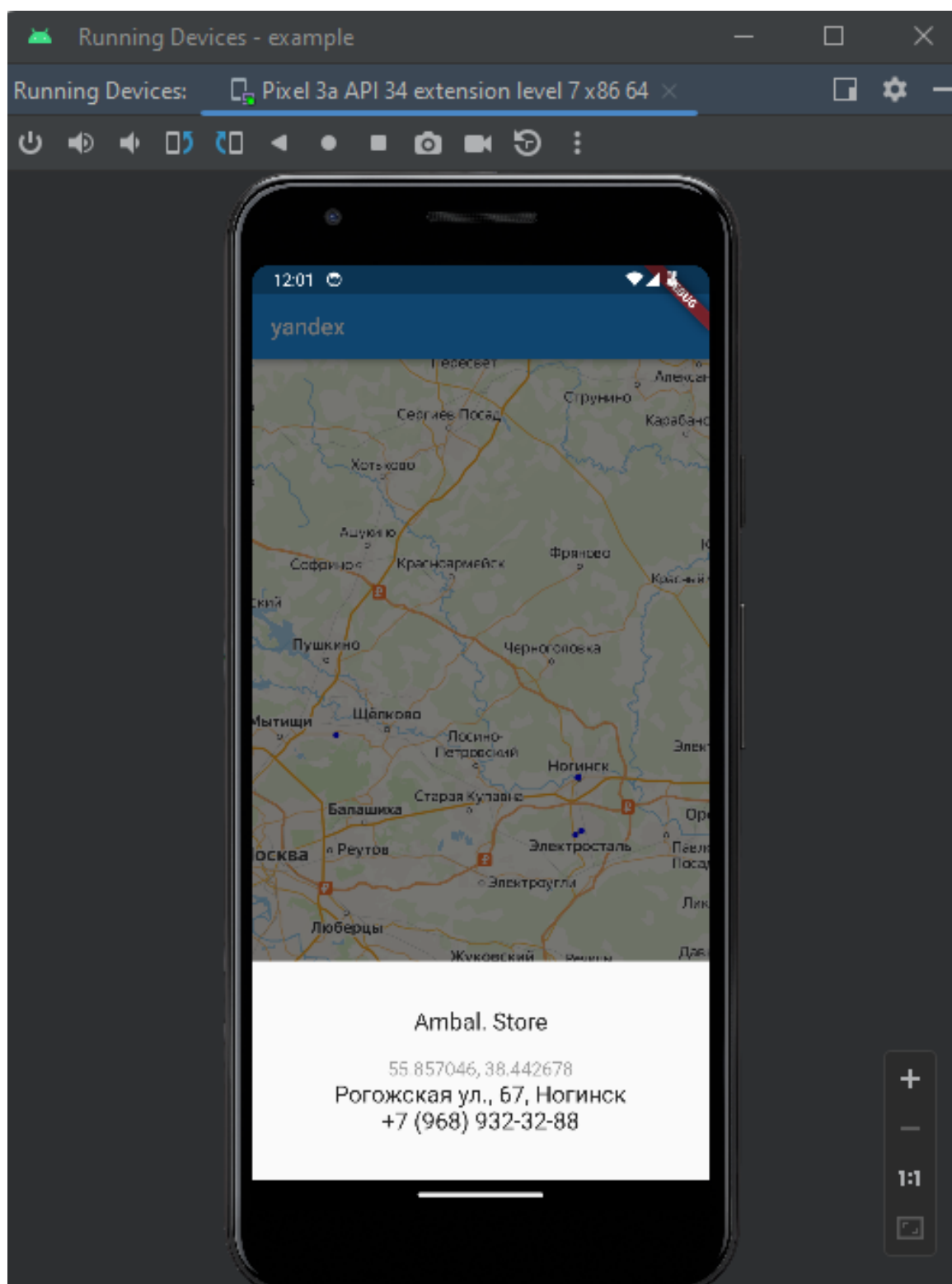


Рис. 3 — Результат работы

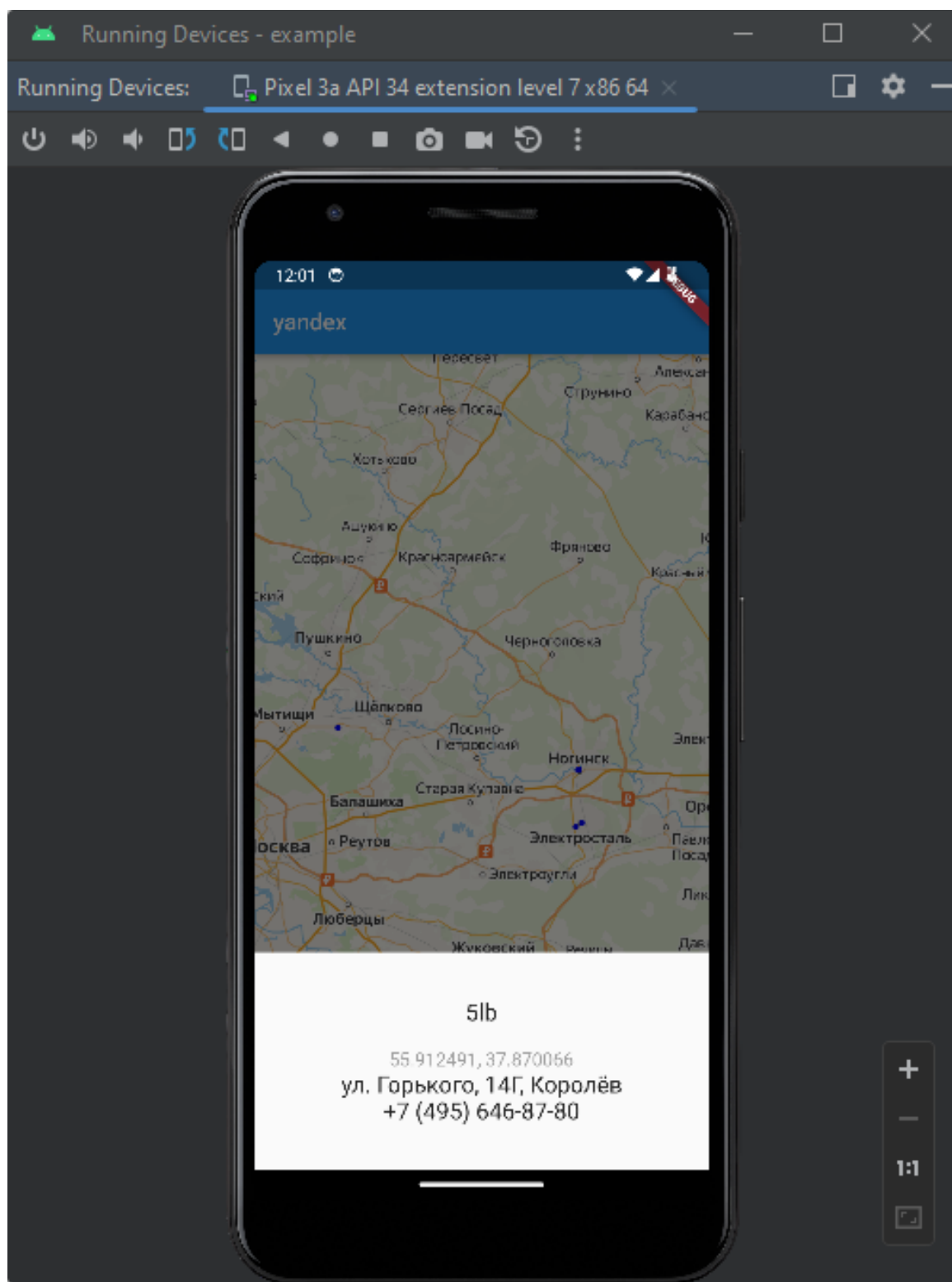


Рис. 4 — Результат работы

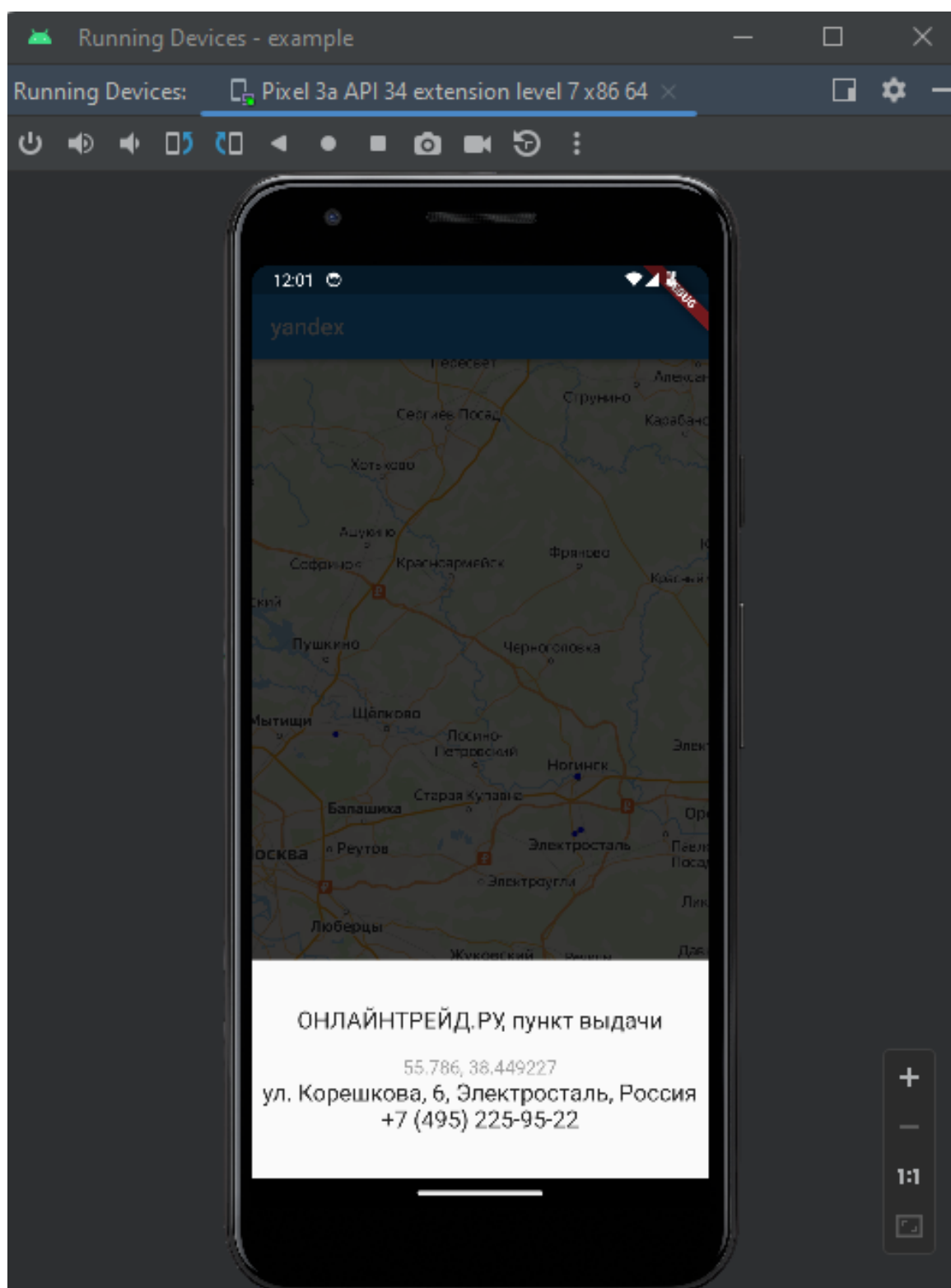


Рис. 5 — Результат работы

5 Выводы

В результате лабораторной работы было создано приложение с графическим пользовательским интерфейсом на языке программирования Dart. Приложение позволяет отображать Яндекс.Карту с расположенными на ней метками объектов, по клику на метку объекта открывается виджет с подробной информацией об объекте. Данные в формате JSON запрашиваются по соответствующему URL, приведенному в таблице вариантов.