|  |  |
| --- | --- |
|  | **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 Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа №7**

по курсу «Разработка мобильных приложений».

«Разработка виджета Yandex maps»

Выполнил:

студент группы ИУ9-71Б

Митрошкин Алексей

Проверил:

Посевин Д.П.

Москва, 2024

1. **Цель**

Разработать виджет Yandex карт.

1. **Задание**

* Подключения Yandex maps API.
* Разработка виджета .

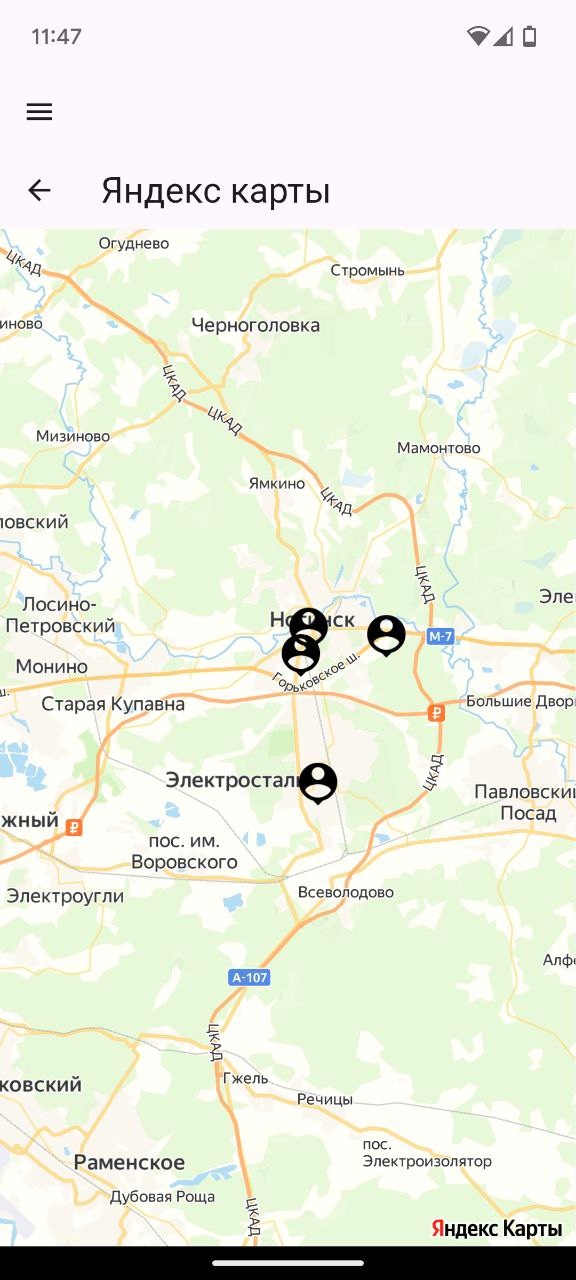
1. **Реализация**

Исходный код приведён в Листинге 1.

Листинг 1

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:yandex\_mapkit/yandex\_mapkit.dart';  import '../types/yandex\_map.dart';  class YandexScreen extends StatefulWidget {    const YandexScreen({super.key});    @override    State<YandexScreen> createState() => \_YandexScreenState();  }  class \_YandexScreenState extends State<YandexScreen> {    late final YandexMapController \_mapController;    @override    void dispose() {      \_mapController.dispose();      super.dispose();    }    @override    Widget build(BuildContext context) {      return Scaffold(        appBar: AppBar(title: const Text('Яндекс карты')),        body: YandexMap(          onMapCreated: (controller) async {             marks = await parsePoints();            \_mapController = controller;            await \_mapController.moveCamera(              CameraUpdate.newCameraPosition(                const CameraPosition(                  target: Point(                    latitude: 50,                    longitude: 20,                  ),                  zoom: 3,                ),              ),            );          }, mapObjects: List.from(generatePlaceMarks(context))        ),      );    }  }  class \_ModalBodyView extends StatelessWidget {    const \_ModalBodyView({required this.point});    final MyPoint point;    @override    Widget build(BuildContext context) {      return Padding(        padding: const EdgeInsets.symmetric(vertical: 100),        child: Column(mainAxisSize: MainAxisSize.min, children: [          Text(point.name, style: const TextStyle(fontSize: 20)),          const SizedBox(height: 20),          Text(            '${point.latitude}, ${point.longitude}',            style: const TextStyle(              fontSize: 16,              color: Colors.blue,            ),          ),        ]),      );    }  }  List<PlacemarkMapObject> generatePlaceMarks(BuildContext context) {    List<PlacemarkMapObject> objects = [];    int i = 0;    for (var point in marks) {      print(point.name);      var newObj = PlacemarkMapObject(mapId: MapObjectId(i.toString()),          point: Point(latitude: point.latitude, longitude: point.longitude),          opacity: 1,          icon: PlacemarkIcon.single(            PlacemarkIconStyle(              image: BitmapDescriptor.fromAssetImage(                'lib/assets/user.png',              ),              scale: 1,            ),          ),          onTap: (\_, \_\_) => showModalBottomSheet(context: context, builder: (context) => \_ModalBodyView(point: point))      );      objects.add(newObj);      i += 1;    }    return objects;  } |

На рисунках 1-2 приведёны результаты работы программы



*Рисунок 1*