**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ**

**Юго-Западный административный округ**

**Государственное бюджетное общеобразовательное учреждение города Москвы**

**«Школа № 1533 «ЛИТ»**

**ВЫПУСКНОЙ ПРОЕКТ**

(специальность «Программирование»)

учащегося группы 10.5 Куликова Дениса Ильясовича

**ГеоАлгоритмические Расчеты: доступность объектов**

Геомаркетинговое приложение для ОС Windows для расчёта и анализа доступности объектов различных типов инфраструктуры

Заказчик: Вилков А. Ю.

Консультанты: Завриев Н.К.

Москва 2020

**Оглавление**

[Введение 3](#_Toc42202451)

[Геоанализ и Геомаркетинг 3](#_Toc42202452)

[Постановка задачи 3](#_Toc42202453)

[Актуальность 4](#_Toc42202454)

[Целевая аудитория 4](#_Toc42202455)

[Решение 5](#_Toc42202456)

[Анализ предметной области 5](#_Toc42202457)

[Программная реализация 6](#_Toc42202458)

[Ход работы 6](#_Toc42202459)

[Результат 9](#_Toc42202460)

[Заключение 12](#_Toc42202461)

[Список литературы 12](#_Toc42202462)

[Приложения 12](#_Toc42202463)

[package.json: 12](#_Toc42202464)

[main.js: 13](#_Toc42202465)

[index.html: 13](#_Toc42202466)

# Введение

## Геоанализ и Геомаркетинг

Геоаналитика - это современный подход в области получения массивов данных о населении от операторов мобильной связи с помощью технологических возможностей «Big Data», с последующим исследованием и статистическим анализом этих данных. На сегодняшний день геоаналитика - единственный источник информации, на основе которого можно исследовать численность и динамику перемещений населения на больших территориях во времени.

Геомаркетинг является одной из многих дисциплин геоаналитики. Геомаркетинг -  технология принятия решений с использованием пространственных данных в процессе планирования и осуществления деятельности в области сбыта продукции, учитывая конкурентную ситуацию и инфраструктуру территории.

Геомаркетинговое исследование поможет оптимально планировать места размещения магазинов торговой сети, находить помещения для коммерческих объектов с учетом доступности, максимального охвата целевой аудитории, конкурентной среды, перспектив развития территорий; понимать наиболее привлекательную ассортиментную программу с учетом потребностей ключевого сегмента аудитории (пешеходный поток), проводить анализ конкурентной среды и прогнозировать размеры товарооборота и объема продаж, а также прочих показателей экономической эффективности присутствия торговой точки в данном конкретном месте.

Сегодня все чаще видно использование геоаналитики для улучшения инфраструктуры городов, а также для развития бизнеса и различных организаций.

# Постановка задачи

Целью моей работы является разработка приложение на ОС Windows для анализа доступности объектов городской инфраструктуры и решения смежных задач, связанных с городской экономикой.

Приложение должно: рассчитывать плотность населения в районах города, определять, насколько выгодно размещать в данной области объекта выбранного типа, определять перспективность размещения в данной области объекта выбранного типа, визуально определять дефицит и профицит объектов различных типов в данной области.

# Актуальность

Очень сложно переоценить умение и возможность использовать геомаркетинговый анализ для успешной борьбы с конкуренцией на рынке. Также не нужно забывать о том, что благодаря геоанализу можно гораздо эффективнее развить бизнес, быстрее увеличить доход компании и успешнее инвестировать в расширение бизнеса.

Области применения геомаркетинга:

* розничная торговля продуктами питания
* здравоохранение (аптеки, частные клиники)
* кафе, рестораны, бары)
* банковский сектор и страхование
* строительные магазины
* торговые центры
* другие сегменты бизнеса, специфика которых - размещение у дома

Типовые задачи геомаркетинга:

* выбор оптимального размещения нового объекта
* оценка целевой аудитории для выбора места под формат магазина, например - доходов
* планирование маркетинговых компаний с учетом размещения клиентов и целевой аудитории
* оценка фактического и потенциального объемов реализации, а также их прогнозирование

# Целевая аудитория

Целевой аудиторией моего приложения являются муниципальные и государственные организации. Они могут использовать данное приложение при планировании развития внутригородских территорий.

Также данное приложение будет полезно бизнесменам и коммерческим организациям для развития и расширения бизнеса.

# Решение

## Анализ предметной области

Геоаналитика - это современный метод в области анализа и визуализации массивов пространственных данных, в том числе с помощью технологических возможностей «Big Data», с последующими выводами и рекомендациями.

На сегодняшний день геоаналитика - основной способ анализа обеспеченности населения городов объектами инфраструктуры.

Геомаркетинг является одним из направлений геоаналитики.

Геомаркетинг -  технология принятия решений с использованием пространственных данных при планировании деятельности в области сбыта продукции, с учетом данных и спросе и конкурентном окружении, а также инфраструктуры территории.

Области применения геомаркетинга:

* розничная торговля
* здравоохранение: аптеки, частные клиники
* общепит: кафе, рестораны, бары
* банковский сектор и страхование
* девелопмент
* торговые центры
* другие направления бизнеса, спецификой которой является –локализованного потребление у дома или работы

Типовые задачи геомаркетинга:

* выбор оптимального размещения нового объекта
* оценка целевой аудитории для выбора места под формат магазина, например – доходов (ценовая ниша)
* планирование маркетинговых компаний с учетом потоков клиентов и целевой аудитории
* оценка фактического и потенциального объемов реализации товаров/работ/услуг, а также их прогнозирование

Геомаркетинговое исследование поможет:

* оптимально спланировать места размещения новых точек продаж или оптимизировать размещение действующих
* найти помещения для бизнеса с учетом фактора их доступности, максимального охвата целевой аудитории, конкурентной среды, перспектив развития территорий;
* понимать наиболее привлекательную ассортиментную линейку с учетом потребностей ключевого сегмента аудитории (пешеходный и/или автопоток)
* проводить анализ конкурентной среды
* прогнозировать размеры товарооборота и объема продаж, а также прочих показателей экономической эффективности присутствия торговой точки в конкретном месте внутригородской территории

## Программная реализация

Для написания данного приложения использовался язык [гипертекстовой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) разметки HTML, язык программирования JavaScript, а также фреймворк для JavaScript Electron JS, благодаря которому стала возможна реализация нативного графического приложения для настольных операционных систем с помощью веб-технологий.

Для отображения карт и векторных слоев на них использовался API 2gis.

Для реализации данной программы использовался [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) Visual Studio Code, а также для работы с таблицами использовалась [программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) для работы с [электронными таблицами](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0) Excel.

## Ход работы

Работа была разбита на несколько частей:

* Редактирование исходных данных
* Создание окна приложения с помощью Electron JS
* Подключение API 2gis и создание карт
* Работа с картами (создание векторных слоев (районы, маркеры, области пешеходной доступности) )
* Визуальное редактирование приложения (добавление подписей, работа с блоками)

Вначале редактировались, форматировались, и разбивались на отдельные файлы данные 2gis, которые были предоставлены заказчиком.

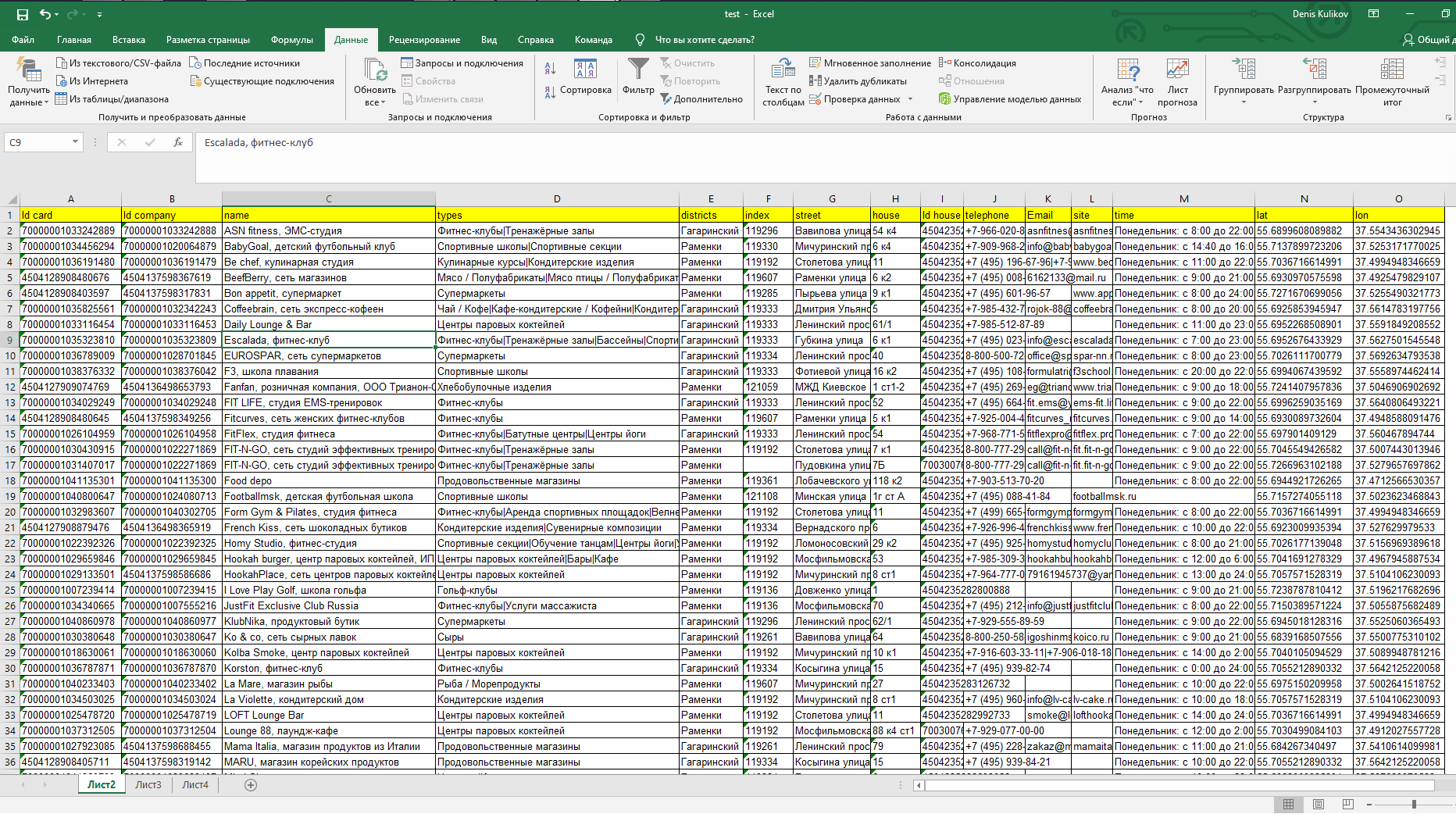


Рисунок 1 До редактирования данных

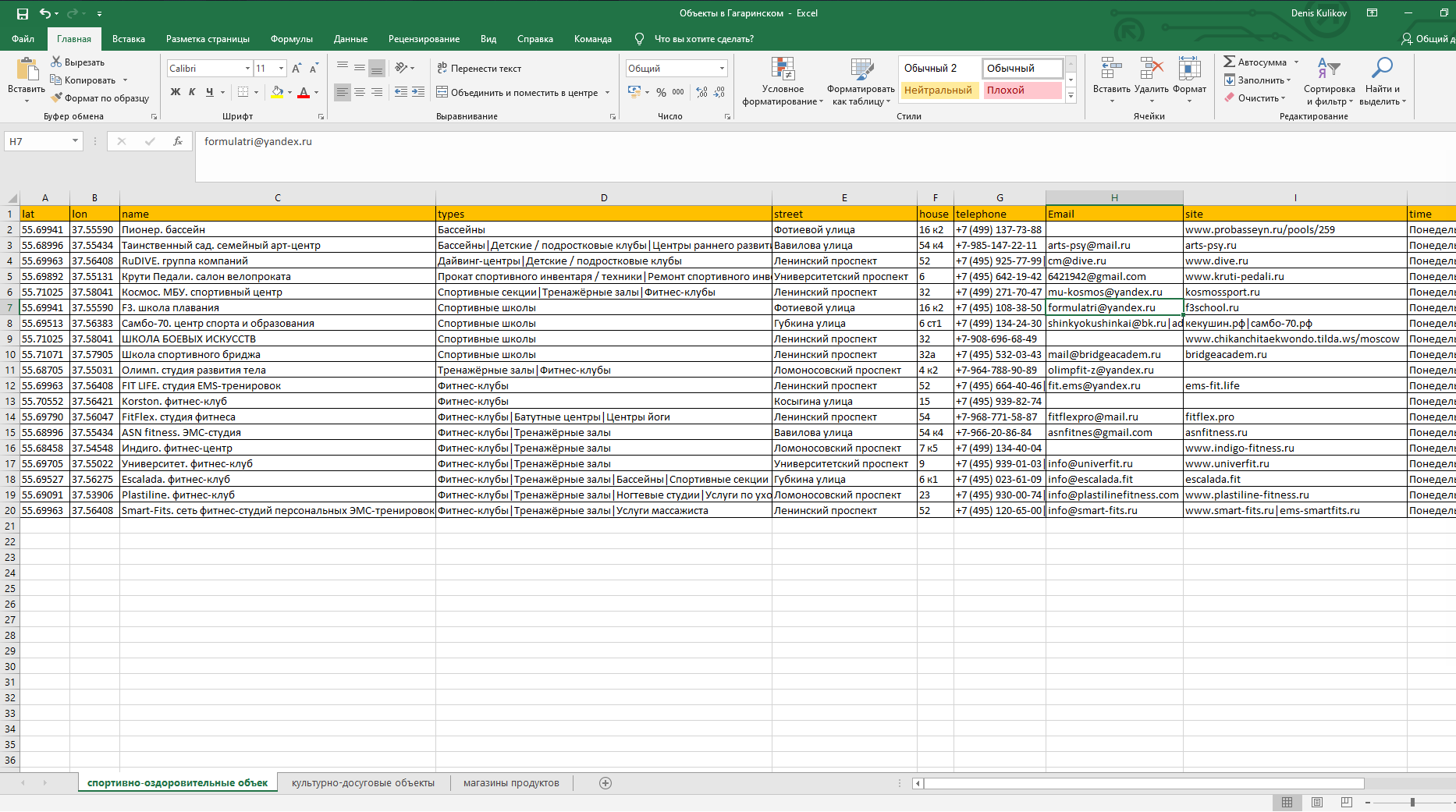


Рисунок 2 После редактирования данных

Затем, используя полученные и отредактированные данные и API 2gis, создавалось готовое приложение.

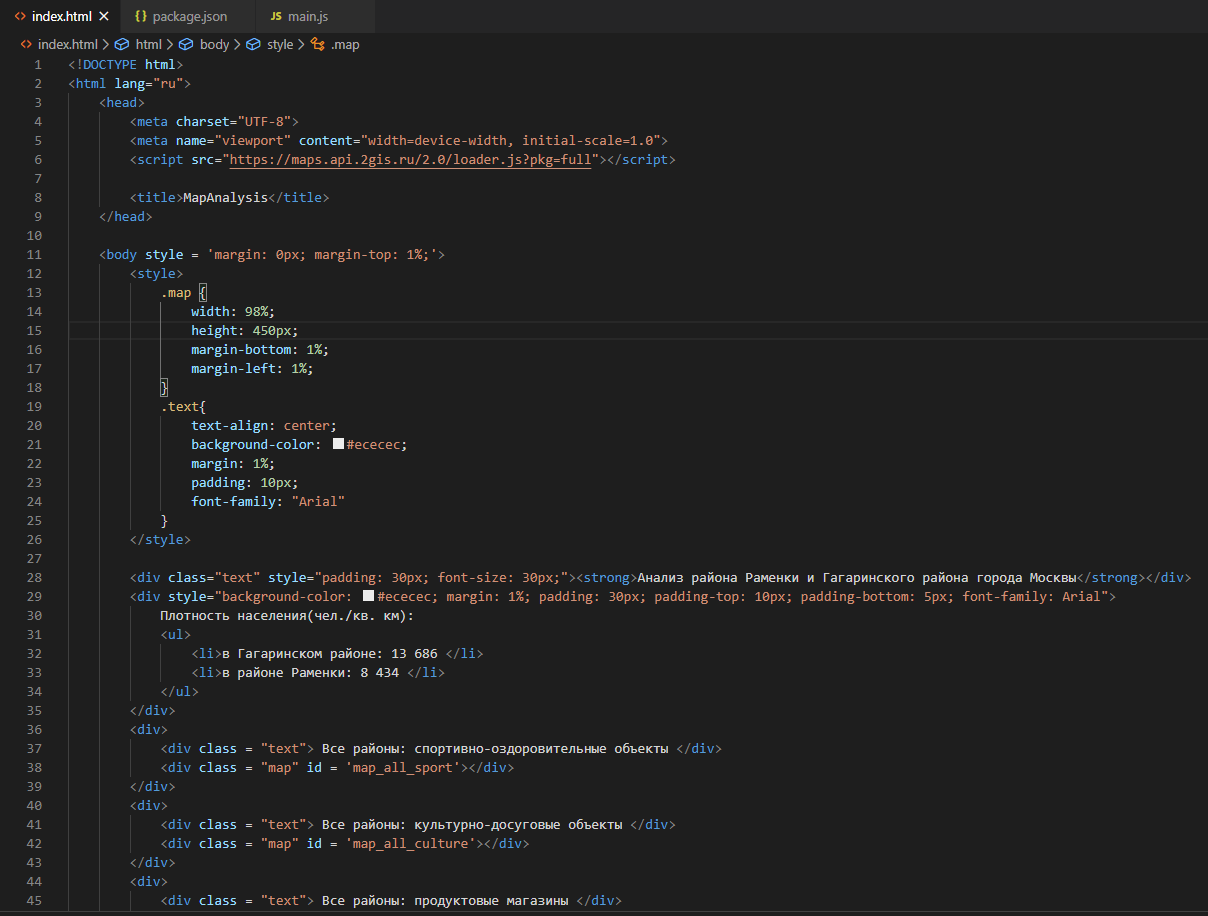


Рисунок 3 Разработка приложения

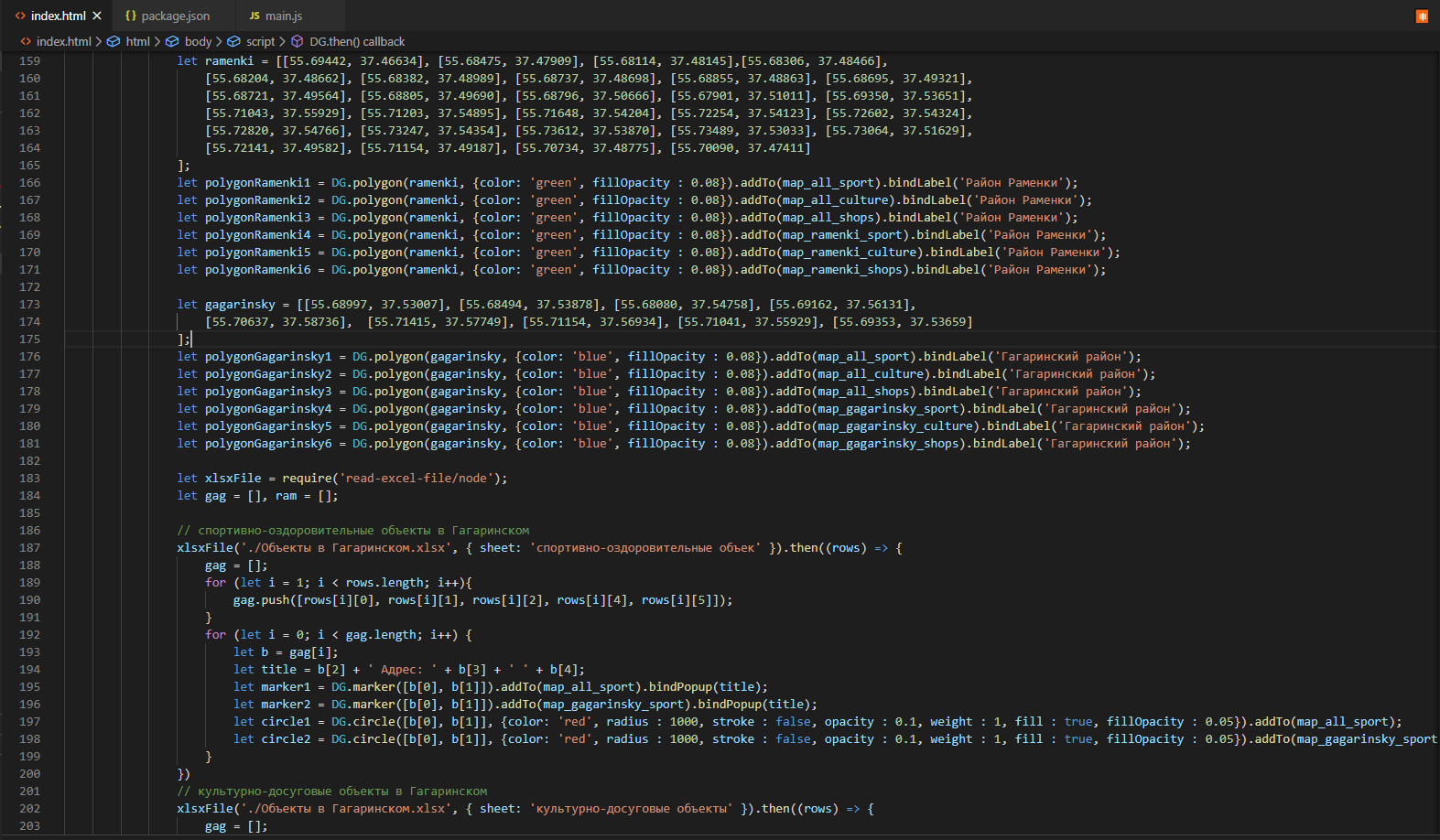


Рисунок 4 Разработка приложения

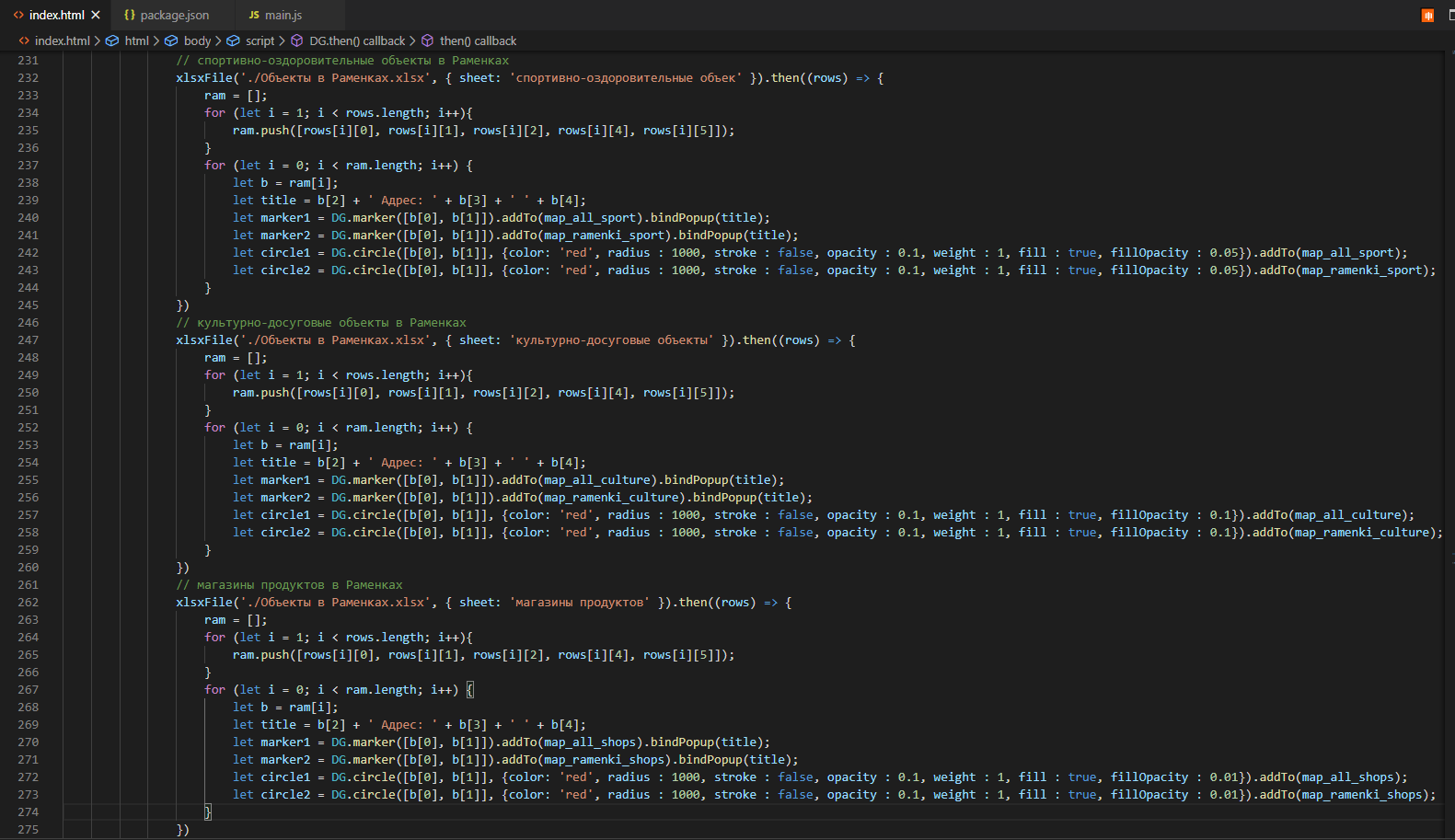


Рисунок 5 Разработка приложения

# Результат

В итоге разработки получилось рабочее приложение, которое отображает объекты разных типов, а также визуально определяет дефицит и профицит объектов разных типов в выбранной области.



Рисунок 6 Готовое приложение

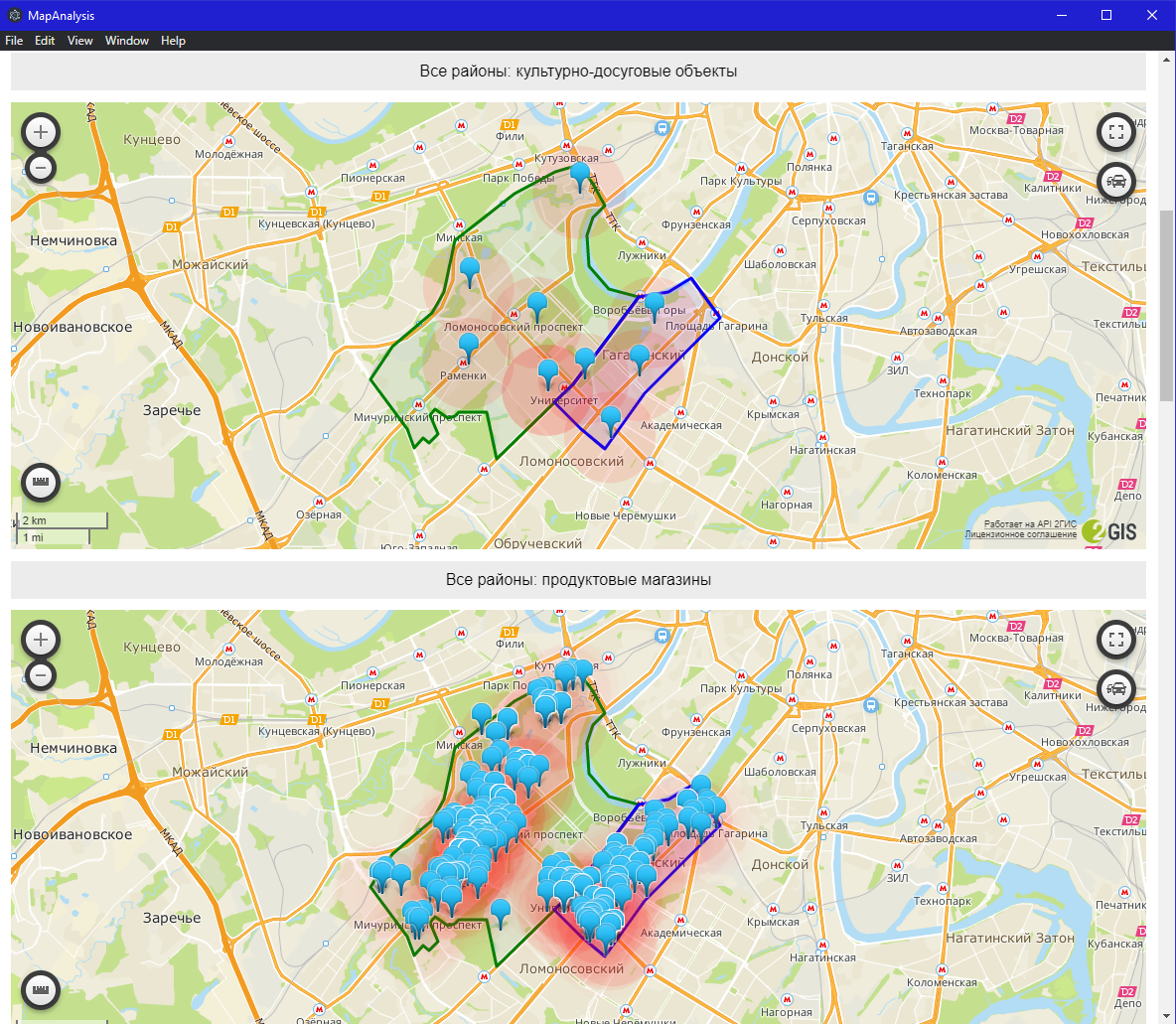


Рисунок 7 Готовое приложение

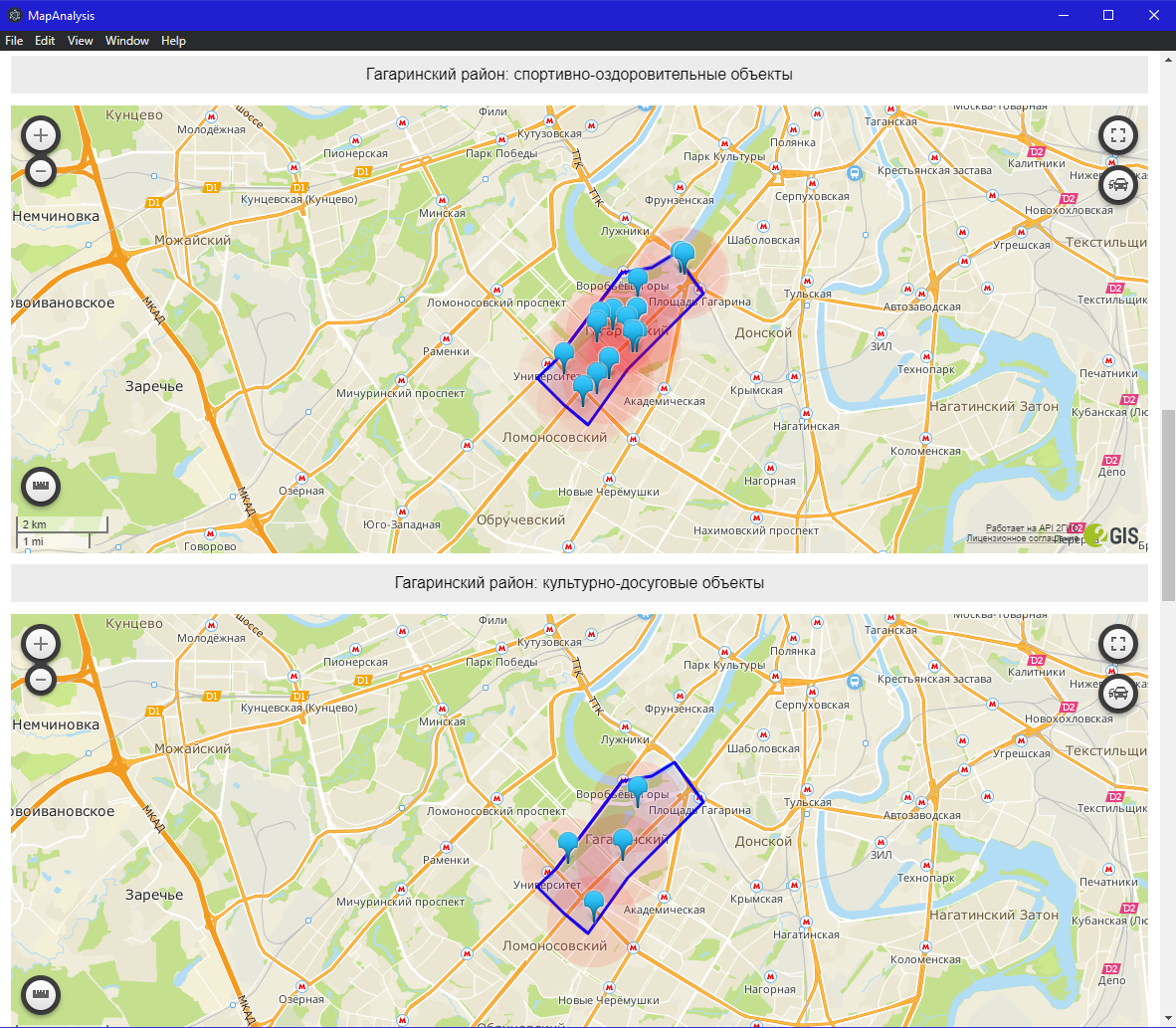


Рисунок 8 Готовое приложение

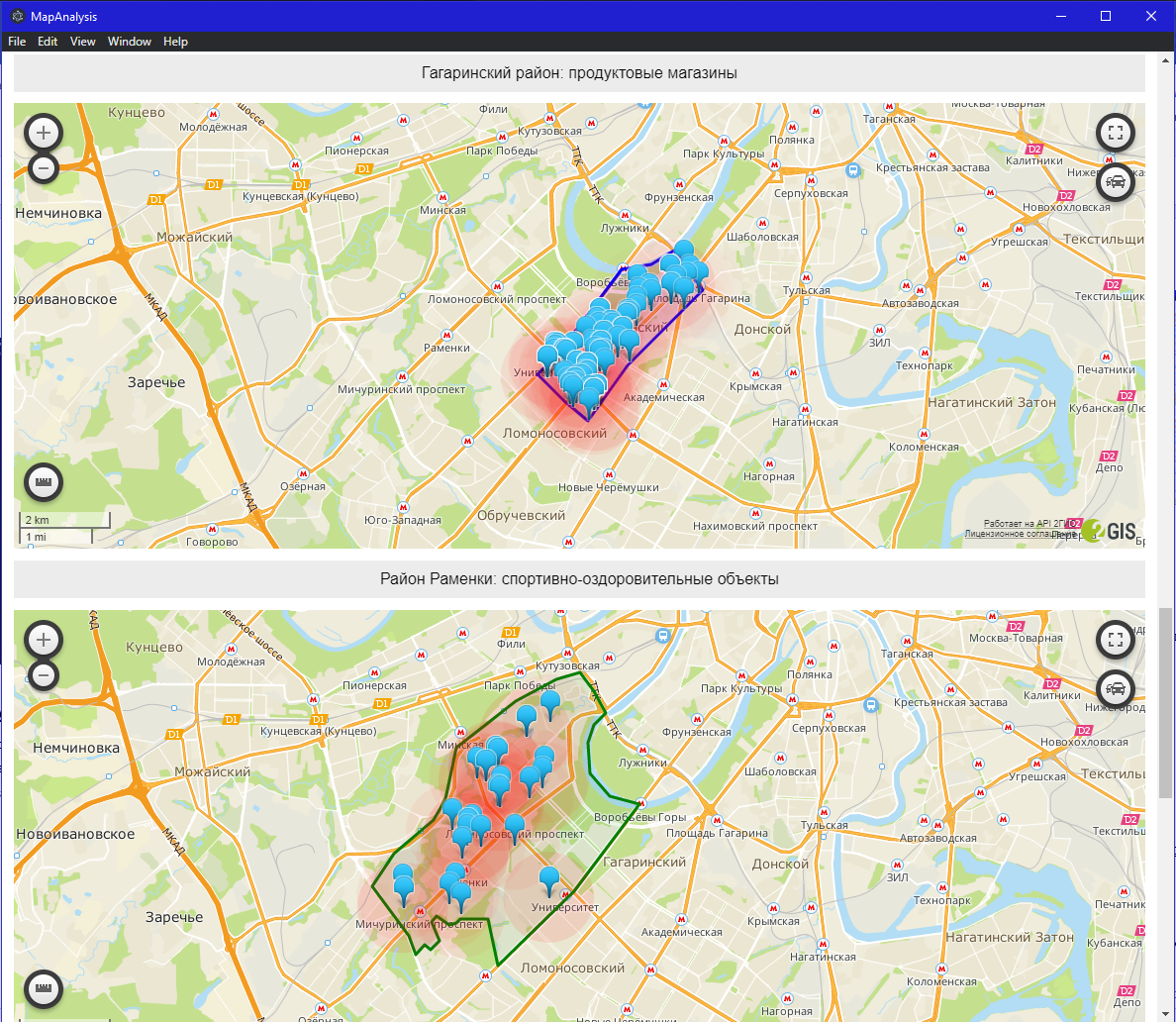


Рисунок 9 Готовое приложение

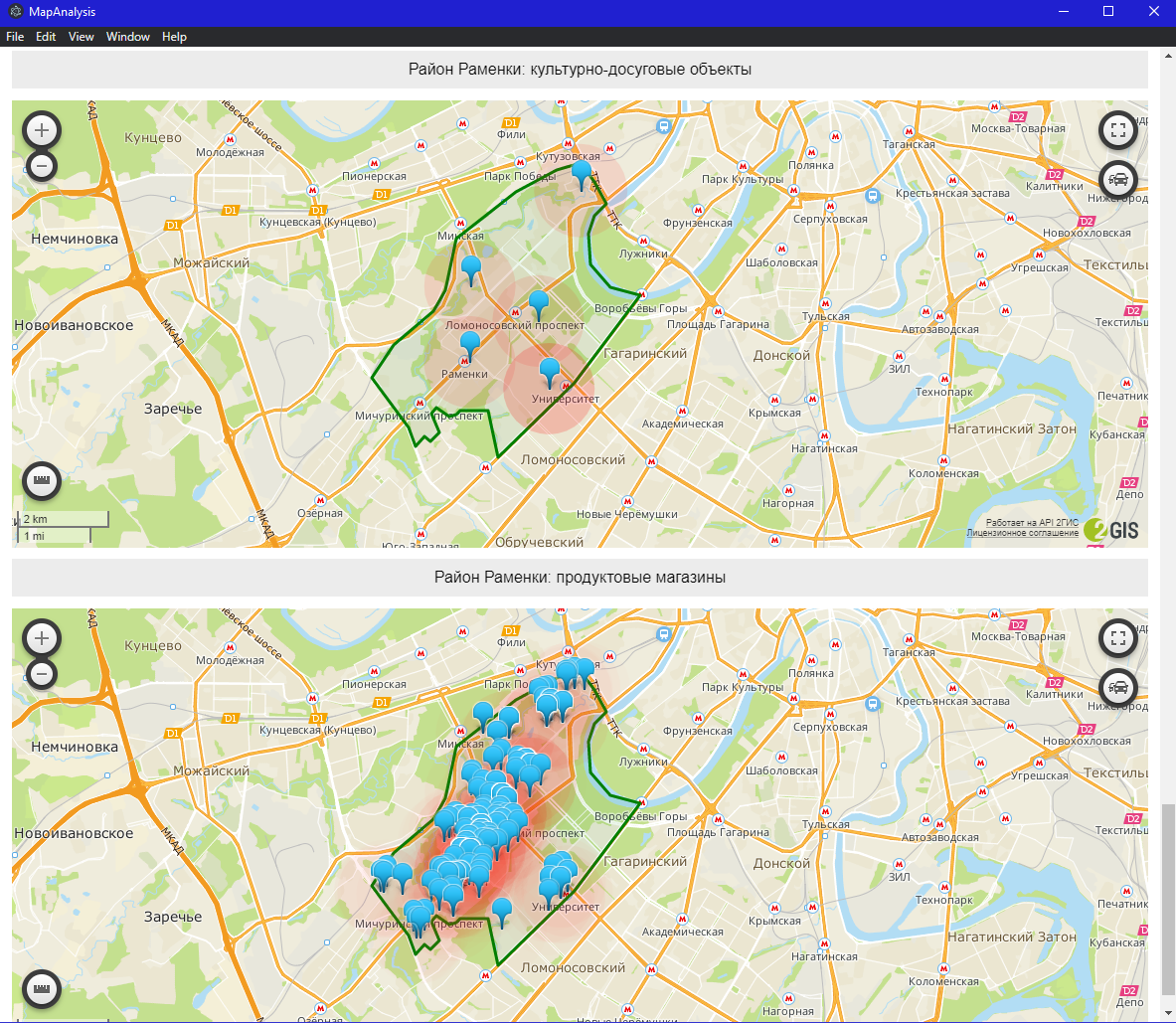


Рисунок 10 Готовое приложение

# Заключение

Задача, поставленная заказчиком, выполнена.

В дальнейшем, данное приложение может улучшаться за счет увеличения анализируемых районов, городов и типов объектов.

Также возможна реализация геоалгоритмических методов, благодаря чему возможен не только визуальный, но и программный анализ данных.

Более того, в будущем желательно создание базы данных, что упростит работу с данными, а также их анализ.

# Список литературы

* Документация к API 2gis [Электронный ресурс]: Режим доступа: <https://api.2gis.ru/doc/maps/ru/quickstart/> (дата последнего обращения – 02.06.2020)
* Документация к Electron JS [Электронный ресурс]: Режим доступа: <https://www.electronjs.org/docs> (дата последнего обращения – 29.05.2020)
* Habr (Electron: разработка настольных приложений с использованием HTML, CSS и JavaScript) [Электронный ресурс]: Режим доступа:<https://habr.com/ru/company/ruvds/blog/436466/> (дата последнего обращения – 25.05.2020)

# Приложения

## package.json:

{

"name": "test",

"version": "1.0.0",

"description": "",

"main": "main.js",

"scripts": {

"start": "electron ."

},

"author": "",

"license": "ISC",

"dependencies": {

"ejs": "^3.1.3",

"electron": "^8.3.0",

"express": "^4.17.1",

"read-excel-file": "^4.0.6"

},

"devDependencies": {

"pug": "^3.0.0"

}

}

## main.js:

const path = require('path');

const url = require('url');

const {app, BrowserWindow} = require('electron');

app.allowRendererProcessReuse = true;

function createWindow(){

let win = new BrowserWindow({

width : 1200,

height : 1060,

minWidth : 1200,

minHeight : 1060,

maxHeight : 1060,

webPreferences: {nodeIntegration: true}

});

win.loadURL(url.format({

pathname : path.join(\_\_dirname, 'index.html'),

protocol : 'file',

slashes : true

}));

win.on('closed', () => {

win = null;

});

}

app.on('ready', createWindow);

## index.html:

<!DOCTYPE html>

<html lang="ru">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <script src="https://maps.api.2gis.ru/2.0/loader.js?pkg=full"></script>

        <title>MapAnalysis</title>

    </head>

    <body style = 'margin: 0px; margin-top: 1%;'>

        <style>

            .map {

                width: 98%;

                height: 450px;

                margin-bottom: 1%;

                margin-left: 1%;

            }

            .text{

                text-align: center;

                background-color: #ececec;

                margin: 1%;

                padding: 10px;

                font-family: "Arial"

            }

        </style>

        <div class="text" style="padding: 30px; font-size: 30px;"><strong>Анализ района Раменки и Гагаринского района города Москвы</strong></div>

        <div style="background-color: #ececec; margin: 1%; padding: 30px; padding-top: 10px; padding-bottom: 5px; font-family: Arial">

            Плотность населения(чел./кв. км):

            <ul>

                <li>в Гагаринском районе: 13 686 </li>

                <li>в районе Раменки: 8 434 </li>

            </ul>

        </div>

        <div>

            <div class = "text"> Все районы: спортивно-оздоровительные объекты </div>

            <div class = "map" id = 'map\_all\_sport'></div>

        </div>

        <div>

            <div class = "text"> Все районы: культурно-досуговые объекты </div>

            <div class = "map" id = 'map\_all\_culture'></div>

        </div>

        <div>

            <div class = "text"> Все районы: продуктовые магазины </div>

            <div class = "map" id = 'map\_all\_shops'></div>

        </div>

        <div>

            <div class = "text"> Гагаринский район: спортивно-оздоровительные объекты </div>

            <div class = "map" id = 'map\_gagarinsky\_sport'></div>

        </div>

        <div>

            <div class = "text"> Гагаринский район: культурно-досуговые объекты </div>

            <div class = "map" id = 'map\_gagarinsky\_culture'></div>

        </div>

        <div>

            <div class = "text"> Гагаринский район: продуктовые магазины </div>

            <div class = "map" id = 'map\_gagarinsky\_shops'></div>

        </div>

        <div>

            <div class = "text"> Район Раменки: спортивно-оздоровительные объекты </div>

            <div class = "map" id = 'map\_ramenki\_sport'></div>

        </div>

        <div>

            <div class = "text"> Район Раменки: культурно-досуговые объекты </div>

            <div class = "map" id = 'map\_ramenki\_culture'></div>

        </div>

        <div>

            <div class = "text"> Район Раменки: продуктовые магазины  </div>

            <div class = "map" id = 'map\_ramenki\_shops'></div>

        </div>

        <script type = "text/javascript">

            DG.then(function () {

                let map\_all\_sport = DG.map('map\_all\_sport', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_all\_culture = DG.map('map\_all\_culture', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_all\_shops = DG.map('map\_all\_shops', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_gagarinsky\_sport = DG.map('map\_gagarinsky\_sport', {

                    center: [55.699549, 37.544726],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_gagarinsky\_culture = DG.map('map\_gagarinsky\_culture', {

                    center: [55.699549, 37.544726],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_gagarinsky\_shops = DG.map('map\_gagarinsky\_shops', {

                    center: [55.699549, 37.544726],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_ramenki\_sport = DG.map('map\_ramenki\_sport', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_ramenki\_culture = DG.map('map\_ramenki\_culture', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                let map\_ramenki\_shops = DG.map('map\_ramenki\_shops', {

                    center: [55.704919, 37.538373],

                    zoom: 12,

                    minZoom: 11,

                    poi : false

                });

                DG.control.scale().addTo(map\_all\_sport);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_all\_sport);

                DG.control.traffic().addTo(map\_all\_sport);

                DG.control.scale().addTo(map\_all\_culture);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_all\_culture);

                DG.control.traffic().addTo(map\_all\_culture);

                DG.control.scale().addTo(map\_all\_shops);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_all\_shops);

                DG.control.traffic().addTo(map\_all\_shops);

                DG.control.scale().addTo(map\_gagarinsky\_sport);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_gagarinsky\_sport);

                DG.control.traffic().addTo(map\_gagarinsky\_sport);

                DG.control.scale().addTo(map\_gagarinsky\_culture);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_gagarinsky\_culture);

                DG.control.traffic().addTo(map\_gagarinsky\_culture);

                DG.control.scale().addTo(map\_gagarinsky\_shops);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_gagarinsky\_shops);

                DG.control.traffic().addTo(map\_gagarinsky\_shops);

                DG.control.scale().addTo(map\_ramenki\_sport);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_ramenki\_sport);

                DG.control.traffic().addTo(map\_ramenki\_sport);

                DG.control.scale().addTo(map\_ramenki\_culture);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_ramenki\_culture);

                DG.control.traffic().addTo(map\_ramenki\_culture);

                DG.control.scale().addTo(map\_ramenki\_shops);

                DG.control.ruler({position: 'bottomleft'}).addTo(map\_ramenki\_shops);

                DG.control.traffic().addTo(map\_ramenki\_shops);

                let ramenki = [[55.69442, 37.46634], [55.68475, 37.47909], [55.68114, 37.48145],[55.68306, 37.48466],

                    [55.68204, 37.48662], [55.68382, 37.48989], [55.68737, 37.48698], [55.68855, 37.48863], [55.68695, 37.49321],

                    [55.68721, 37.49564], [55.68805, 37.49690], [55.68796, 37.50666], [55.67901, 37.51011], [55.69350, 37.53651],

                    [55.71043, 37.55929], [55.71203, 37.54895], [55.71648, 37.54204], [55.72254, 37.54123], [55.72602, 37.54324],

                    [55.72820, 37.54766], [55.73247, 37.54354], [55.73612, 37.53870], [55.73489, 37.53033], [55.73064, 37.51629],

                    [55.72141, 37.49582], [55.71154, 37.49187], [55.70734, 37.48775], [55.70090, 37.47411]

                ];

                let polygonRamenki1 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_all\_sport).bindLabel('Район Раменки');

                let polygonRamenki2 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_all\_culture).bindLabel('Район Раменки');

                let polygonRamenki3 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_all\_shops).bindLabel('Район Раменки');

                let polygonRamenki4 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_ramenki\_sport).bindLabel('Район Раменки');

                let polygonRamenki5 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_ramenki\_culture).bindLabel('Район Раменки');

                let polygonRamenki6 = DG.polygon(ramenki, {color: 'green', fillOpacity : 0.08}).addTo(map\_ramenki\_shops).bindLabel('Район Раменки');

                let gagarinsky = [[55.68997, 37.53007], [55.68494, 37.53878], [55.68080, 37.54758], [55.69162, 37.56131],

                    [55.70637, 37.58736],  [55.71415, 37.57749], [55.71154, 37.56934], [55.71041, 37.55929], [55.69353, 37.53659]

                ];

                let polygonGagarinsky1 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_all\_sport).bindLabel('Гагаринский район');

                let polygonGagarinsky2 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_all\_culture).bindLabel('Гагаринский район');

                let polygonGagarinsky3 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_all\_shops).bindLabel('Гагаринский район');

                let polygonGagarinsky4 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_gagarinsky\_sport).bindLabel('Гагаринский район');

                let polygonGagarinsky5 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_gagarinsky\_culture).bindLabel('Гагаринский район');

                let polygonGagarinsky6 = DG.polygon(gagarinsky, {color: 'blue', fillOpacity : 0.08}).addTo(map\_gagarinsky\_shops).bindLabel('Гагаринский район');

                let xlsxFile = require('read-excel-file/node');

                let gag = [], ram = [];

                // спортивно-оздоровительные объекты в Гагаринском

                xlsxFile('./Объекты в Гагаринском.xlsx', { sheet: 'спортивно-оздоровительные объек' }).then((rows) => {

                    gag = [];

                    for (let i = 1; i < rows.length; i++){

                        gag.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < gag.length; i++) {

                        let b = gag[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_sport).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_gagarinsky\_sport).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.05}).addTo(map\_all\_sport);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.05}).addTo(map\_gagarinsky\_sport);

                    }

                })

                // культурно-досуговые объекты в Гагаринском

                xlsxFile('./Объекты в Гагаринском.xlsx', { sheet: 'культурно-досуговые объекты' }).then((rows) => {

                    gag = [];

                    for (let i = 1; i < rows.length; i++){

                        gag.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < gag.length; i++) {

                        let b = gag[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_culture).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_gagarinsky\_culture).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.1}).addTo(map\_all\_culture);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.1}).addTo(map\_gagarinsky\_culture);

                    }

                })

                // магазины продуктов в Гагаринском

                xlsxFile('./Объекты в Гагаринском.xlsx', { sheet: 'магазины продуктов' }).then((rows) => {

                    gag = [];

                    for (let i = 1; i < rows.length; i++){

                        gag.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < gag.length; i++) {

                        let b = gag[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_shops).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_gagarinsky\_shops).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.01}).addTo(map\_all\_shops);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.01}).addTo(map\_gagarinsky\_shops);

                    }

                })

                // спортивно-оздоровительные объекты в Раменках

                xlsxFile('./Объекты в Раменках.xlsx', { sheet: 'спортивно-оздоровительные объек' }).then((rows) => {

                    ram = [];

                    for (let i = 1; i < rows.length; i++){

                        ram.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < ram.length; i++) {

                        let b = ram[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_sport).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_ramenki\_sport).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.05}).addTo(map\_all\_sport);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.05}).addTo(map\_ramenki\_sport);

                    }

                })

                // культурно-досуговые объекты в Раменках

                xlsxFile('./Объекты в Раменках.xlsx', { sheet: 'культурно-досуговые объекты' }).then((rows) => {

                    ram = [];

                    for (let i = 1; i < rows.length; i++){

                        ram.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < ram.length; i++) {

                        let b = ram[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_culture).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_ramenki\_culture).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.1}).addTo(map\_all\_culture);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.1}).addTo(map\_ramenki\_culture);

                    }

                })

                // магазины продуктов в Раменках

                xlsxFile('./Объекты в Раменках.xlsx', { sheet: 'магазины продуктов' }).then((rows) => {

                    ram = [];

                    for (let i = 1; i < rows.length; i++){

                        ram.push([rows[i][0], rows[i][1], rows[i][2], rows[i][4], rows[i][5]]);

                    }

                    for (let i = 0; i < ram.length; i++) {

                        let b = ram[i];

                        let title = b[2] + ' Адрес: ' + b[3] + ' ' + b[4];

                        let marker1 = DG.marker([b[0], b[1]]).addTo(map\_all\_shops).bindPopup(title);

                        let marker2 = DG.marker([b[0], b[1]]).addTo(map\_ramenki\_shops).bindPopup(title);

                        let circle1 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.01}).addTo(map\_all\_shops);

                        let circle2 = DG.circle([b[0], b[1]], {color: 'red', radius : 1000, stroke : false, opacity : 0.1, weight : 1, fill : true, fillOpacity : 0.01}).addTo(map\_ramenki\_shops);

                    }

                })

            });

        </script>

    </body>

</html>