

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Разработка программного обеспечения информационных
систем»

Тема: Построение маршрутов - Mongo

Студентка гр. 5303	_____	Клименко А.Ю.
Студентка гр. 5303	_____	Федорова М.Д.
Студент гр. 5303	_____	Шабашов Н.А.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург
2018

ЗАДАНИЕ НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты Клименко А.Ю., Федорова М.Д., Шабашов Н.А.

Группа 5303

Тема проекта: Построение маршрутов - Mongo

Исходные данные:

Проект должен быть разработан с использованием базы данных Mongo

Содержание пояснительной записки:

Содержание, Введение, Качественные требования к решению, Сценарии использования, Модель данных, Разработанное приложение, Заключение, Список использованных источников.

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 13.09.2018

Дата сдачи реферата: 27.12.2018

Дата защиты реферата: 27.12.2018

Студентка гр.5303	_____	Клименко А.Ю.
Студентка гр.5303	_____	Федорова М.Д.
Студент гр.5303	_____	Шабашов Н.А.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В ходе выполнения данного индивидуального домашнего задания был реализован веб-сервис на основе СУБД MongoDB, с помощью которого выполняются следующие функции:

- построение маршрута из одной точки на карте в другую;
- просмотр информации о заданной точке на карте;
- экспорт данных.

SUMMARY

In the course of this individual homework, a MongoDB-based web service was implemented, with which the following functions are performed:

- building a route from one point on the map to another;
- view information about a given point on the map;
- export data.

СОДЕРЖАНИЕ

Оглавление

Введение.....	5
1. Качественные требования к решению	6
2. Сценарии использования.....	7
2.1. Сценарии использования для задачи импорта, представления, анализа и экспорта данных.....	7
2.2. Вывод	8
3. Модель данных.....	9
3.1. Описание структуры.....	9
3.2. Нереляционная модель данных.....	10
3.3. Аналог модели данных для SQL СУБД.....	11
3.4. Запросы	13
3.5. Выводы.....	14
4. Разработанное приложение	15
4.1. Краткое описание.....	15
4.2. Используемые технологии	15
4.3. Ссылки на Приложение.....	16
Список использованных источников	18
Приложение А. Документация по сборке и развертыванию приложения	19
Приложение В. Инструкция для пользователя	20
Приложение С. Снимки экрана приложения	21

ВВЕДЕНИЕ

На данный момент в России насчитывается более 40,9 млн легковых автомобилей. В среднем на каждую тысячу жителей приходится 284 автомобиля. Это означает, что потребность просмотра маршрута из одной точки на карте в другую возрастает.

Целью проекта является разработка приложения, с помощью которого можно построить маршрут из одной точки на карте в другую.

В проекте разработано веб-приложение на основе СУБД MongoDB.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Необходимо разработать веб-приложение, позволяющее построить маршрут из одной точки на карте в другую.

Основные функции:

- Поиск и просмотр информации по названию/адресу;
- Информация по координате;
- Построение маршрутов;
- Экспорт данных.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Сценарии использования для задачи импорта, представления, анализа и экспорта данных

Для экспорта данных необходимо отправить rest-запрос к одной из коллекций. Примеры запросов представлены ниже:

`http://localhost:8090/api/way/export?limit=50&skip=0`

`http://localhost:8090/api/tips/export?limit=50&skip=0`

`http://localhost:8090/api/route/node/export?limit=50&skip=0`

`http://localhost:8090/api/route/export?limit=50&skip=0`

<http://localhost:8090/api/node/export?limit=50&skip=0>

Здесь:

way – название коллекции;

export – операция;

limit=50 – ограничение на вывод.



```
[{"name": "Шепетовская улица", "wayId": 4078548}, {"name": "Лодейное Поле — Вытегра", "wayId": 4397228}, {"name": "«Кола»", "wayId": 4411466}, {"name": "Свирское - Горка", "wayId": 4411514}, {"name": "Лодейное Поле — Вытегра", "wayId": 4411527}, {"name": "Лодейное Поле — Вытегра", "wayId": 4411539}, {"name": "Мурманское шоссе", "wayId": 4411560}, {"name": "Мурманское шоссе", "wayId": 4411561}, {"name": "Подъезд к г- Лодейное Поле", "wayId": 4411575}, {"name": "подъезд к деревне Кондуши", "wayId": 4411580}, {"name": "подъезд к деревне Кондуши", "wayId": 4411581}, {"name": "улица Гагарина", "wayId": 4414861}, {"name": "Казанский переулок", "wayId": 4414865}, {"name": "бульвар Алексея Толстого", "wayId": 4414959}, {"name": "Ахматовская улица", "wayId": 4414960}, {"name": "Садовая улица", "wayId": 4414988}, {"name": "Парашютная улица", "wayId": 4454283}, {"name": "Калинин переулок", "wayId": 4454291}, {"name": "Екатерининский проспект", "wayId": 4454334}, {"name": "Виндавская улица", "wayId": 4454357}, {"name": "Зоологический переулок", "wayId": 4454376}, {"name": "набережная Обводного канала", "wayId": 4454381}, {"name": "Коннолахтинская дорога", "wayId": 4454388}, {"name": "КАД", "wayId": 4454389}, {"name": "Кожевенная линия", "wayId": 4454390}, {"name": "площадь Военных Медиков", "wayId": 4454465}, {"name": "Вокзальная улица", "wayId": 4454470}, {"name": "набережная канала Грибоедова", "wayId": 4454526}, {"name": "улица Чернова", "wayId": 4454554}, {"name": "набережная Лебяжьей канавки", "wayId": 4454562}, {"name": "проспект Металлистов", "wayId": 4454579}, {"name": "набережная 30-го Гвардейского Корпуса", "wayId": 4454632}, {"name": "проспект Энергетиков", "wayId": 4454660}, {"name": "улица Севастьянова", "wayId": 4454679}, {"name": "Петергофская улица", "wayId": 4454689}, {"name": "Бородинская улица", "wayId": 4454703}, {"name": "Моховая улица", "wayId": 4454742}, {"name": "Заречная улица", "wayId": 4454755}, {"name": "переулок Гривцова", "wayId": 4454822}, {"name": "Батальонная улица", "wayId": 4454851}, {"name": "улица Александра Невского", "wayId": 4454951}, {"name": "КАД", "wayId": 4455061}, {"name": "улица Доблестин", "wayId": 4455064}, {"name": "улица Нахимова", "wayId": 4455074}, {"name": "Черкасовская автотрасса", "wayId": 4455168}, {"name": "проспект Будённого", "wayId": 4455228}, {"name": "проспект Стачек", "wayId": 4455242}, {"name": "Краснопутиловская улица", "wayId": 4455257}, {"name": "Люблинский переулок", "wayId": 4455260}, {"name": "Хабаровская улица", "wayId": 4455308}]
```

1. «Построение маршрутов»

- 1) Пользователь в поле поиска вводит необходимый адрес/название места и нажимает на кнопку «Найти»;
- 2) Система отмечает на карте найденное место, показывая о нем краткую информацию.
- 3) Пользователь нажимает на кнопку «Проложить маршрут»;

- 4) Система выводит окно «Откуда – куда», где пользователь может ввести место отправления, а место назначения – это уже отмеченное место на карте
- 5) Пользователь вводит адрес отправления и нажимает кнопку «Построить»
- 6) Система отмечает адрес отправления, строит маршрут и выводит его на карту, отображая информацию о расстоянии пути и времени его прохождения.

2. «Информация по координате»

- 1) Пользователь в поле поиска вводит необходимые координаты и нажимает на кнопку «Найти» или отмечает искомое место на карте, кликая на него;
- 2) Система отмечает на карте найденное место, показывая о нем краткую информацию (координаты).

3. «Поиск по названию/адресу»

- 1) Пользователь в поле поиска вводит необходимый адрес/название места и нажимает на кнопку «Найти»;
- 2) Система отмечает на карте найденное место, показывая о нем краткую информацию.

2.2. Вывод

Для решения преобладают операции чтения, так как для Пользователя реализован поиск объектов на карте по адресу/координате и отображение маршрутов.

3. МОДЕЛЬ ДАННЫХ

Модель данных для нереляционной базы данных.

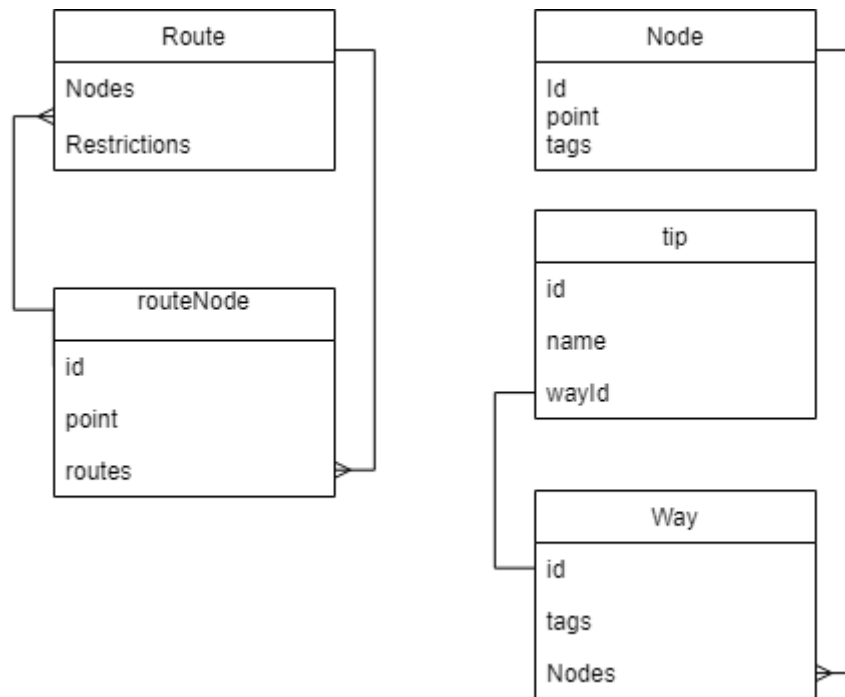


Рисунок 1 – Модель данных нереляционной БД

3.1. Описание структуры

Коллекция Node:

Id – идентификатор точки (int64);

tags – поля точки (массив пар string).

Коллекция Route:

Id – идентификатор маршрута (int64)

Restrictions – ограничения на маршруте (массив int)

Tags – поля маршрута (массив пар string)

Nodes – массив точек

Коллекция routeNodes:

id – идентификатор (int64)

point – координаты точки (2*double)

routes – массив точек(string+int)

Коллекция tip:

id – идентификатор (int64)

name – адрес (string)

wayId – идентификатор пути (int64)

Коллекция Way:

Id – идентификатор пути(int64)

Tags – поля маршрута (массив пар string)

Nodes – массив точек

3.2. Нереляционная модель данных

Расчет памяти для MongoDB:

Средний размер node: 104байта

Средний размер Route: 481 байт

Средний размер Relation: 200 байт

Средний размер routeNode: 156 байт

Средний размер tip: 116 байт

Средний размер Way: 500 байт

Размер node: 1.2Гб

Размер Route: 0.1Гб

Размер Relation: 12Мб

Размер routeNode: 0.25Гб

Размер tip: 8Мб

Размер Way: 0.7Гб

Общий размер 2.3Гб

3.3. Аналог модели данных для SQL СУБД

На рисунке 1 представлена модель данных для реляционной базы данных.

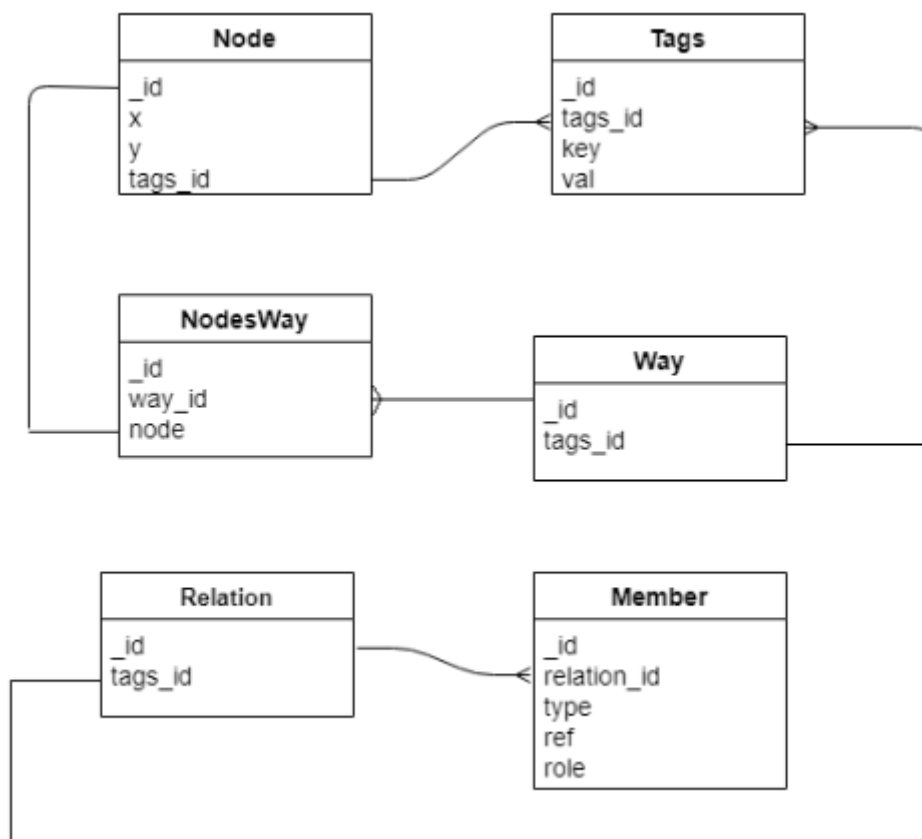


Рисунок 2 – Модель данных реляционной БД

Описание назначений коллекций, типов данных и сущностей

6 коллекций:

1-ая Way - хранит данные о путях.

- id () - идентификатор пути;
- tags id - id тега пути

2-ая Node - хранит точки

- id - идентификатор точки;
- x - координата x
- y - координата y
- tags id - id тега пути

3-я Relation - хранит отношения

- id - идентификатор точки;
- tags id - id тега пути

4-ая Member - хранит данные об участниках

- id - идентификатор участника;
- relation id - идентификатор отношения
- type - хранит тип
- ref - хранит ссылку на id участника отношения
- role - роль участника отношения

5-ая NodesWay

- id - идентификатор
- way id - идентификатор пути;
- node - точки;

6-ая Tags

- id - идентификатор
- tags id - идентификатор тега
- key - ключ тега
- val - значение тега

Оценка удельного объема информации, хранимой в модели

Среднее количество тегов = 100;

Среднее количество nodes = 100

Среднее количество relation = 1000000

Среднее количество member = 10000000

Size (nodes) = 152

Size (way) = 88

Size (NodesWay) = 36

Size (Way) = 24

Size (Relarion) = 24

Size (Member) = 100

$$\text{Size} = (10088 + 12 + 100(88 + 88100) + 100100) * n$$

3.4. Запросы

Примеры запросов представлены в таблице 1.

Таблица 1 – Примеры запросов

Формулировка запроса	MongoDB	SQL
Поиск места по названию	db.getCollection("Tip").find({ "name" : { "\$regex" : "r" , "\$options" : "i" } })	SELECT * From Tip t WHERE t.name == REGEXP_LIKE('.*r.*');
Поиск дорог	db.getCollection("Way").find({ "tags.highway" :"road" } })	SELECT * FROM Way w WHERE w.type == "restriction";
Информация по координатам	db.getCollection('tip').find({ "wayId" : 23744795 })	SELECT * FROM tip t WHERE t.wayId == 23744795;
Ограничения на дорогах	db.getCollection("Relation").find({ "tags.type" :"restriction" } })	SELECT * FROM Relation r WHERE r.type == "restriction";
Ближайшую к координате	db.getCollection("Node").find({ "point" : {	SELECT * FROM Node n WHERE n.x > 55.7 AND n.x < 55.8 AND n.y > 55.7 AND n.y < 55.8;

	"\$near" : { "x" : 59.9379492371 1271 , "y" : 30.3604715156 15167}}})	
--	---	--

3.5. Выводы

Для рассматриваемой задачи наиболее подходящей базой данных будет нереляционная БД, т.к. сложность запросов будет меньше. Но у нереляционной базы данных тоже есть свои недостатки – объем затрачиваемой памяти больше.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Краткое описание

Разработанное приложение осуществляет построение маршрутов. Приложение состоит из главной станицы, на которой отображаются следующие формы:

1. Форма поиска по адресу/названию;
2. Форма построения маршрутов с информацией откуда и куда будет строиться маршрут;
3. Форма «popUp» с информацией о данной метке на карте.

4.2. Используемые технологии

При написании приложения использовались следующие технологии:

- Java 8 – языки программирования, использованные в написании сервера;
- Spring Data – инструмент для доступа к базе данных MongoDB;
- Spring Boot – фреймворк для ускоренного написания приложений с Rest API;
- MongoDB – документированная база данных, взятая за основу и для изучения в данной работе;
- Maven – это инструмент для сборки Java проекта;
- Spring framework – фреймворк, созданный для облегчения написания веб-приложений. В данном приложении используются модуль Boot для упрощенного развертывания системы, модуль Data для создания упрощенных GET-запросов к базе данных и модель MVC для настройки веб-приложения.
- Apache commons – библиотека для открытия файла с картами;
- Junit - библиотека для модульного тестирования программного обеспечения на языке Java;

- Google code gson – библиотека, позволяющая конвертировать объекты JSON в Java-объекты и наоборот;
- Lombok – это библиотека, с помощью которой можно сократить количество шаблонного кода, который нужно писать на Java;
- Leaflet – библиотека с открытым исходным кодом, написанная на JavaScript, предназначенная для отображения карт на веб-сайтах;
- HTML – стандартизированный язык разметки документов;
- CSS – формальный язык описания внешнего вида документа, написанного с использованием языка разметки;
- Java Script – языки программирования, использованные в написании UI слоя приложения
- JQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML.

4.3. Ссылки на Приложение

Исходный код приложения и инструкция по установке находятся по ссылке:

https://github.com/moevm/nosql2018-mongo_routing

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного индивидуального домашнего задания было разработано приложение с использованием базы данных MongoDB, с помощью которого выполняются следующие функции:

- построение маршрута из одной точки на карте в другую;
- просмотр информации о заданной точке на карте;
- экспорт данных.

Было проведено тестирование разработанного приложения, которое прошло успешно. Приложение работает корректно.

При создании приложения были изучены особенности работы нереляционных баз данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Java Platform Standard Edition 8 Documentation URL:
<https://docs.oracle.com/javase/8/docs/> (дата обращения: 19.12.2018).
2. Welcome to the MongoDBDB Docs. URL:
https://docs.MongoDBdb.com/?_ga=2.140482107.947430166.1545497958-550656450.1544206702 дата обращения: 19.12.2018).
3. Добро пожаловать на OpenStreetMap! URL:
<https://www.openstreetmap.org/#map=16/54.3561/18.6303> (дата обращения: 19.12.2018).

ПРИЛОЖЕНИЕ А. ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ

Инструкция по сборке и запуску:

1. Скачать проект из репозитория.
2. Собрать из него файл с расширением .jar. Это можно сделать с помощью технологии Maven или средствами IDE.
3. Скачать базу данных MongoDB.
4. Запустить базу данных MongoDB.
5. Запустить jar-файл.
6. Перейти в браузере по адресу: <http://localhost:8090>.

ПРИЛОЖЕНИЕ В. ИНСТРУКЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ

При входе на сайт отображается географическая карта.

В левом верхнем углу располагается поиск по карте. Пользователь может ввести название или улицу и, выбрав из предложенных вариантов, посмотреть информацию по конкретному месту.

Просмотреть информацию о месте можно нажав на него. В появившемся окне будут показаны координаты данного места.

Построить маршрут пользователь может, выбрав пункт отправления и пункт назначения.

ПРИЛОЖЕНИЕ С. СНИМКИ ЭКРАНА ПРИЛОЖЕНИЯ

На рисунках 2-5 изображены снимки экрана приложения.

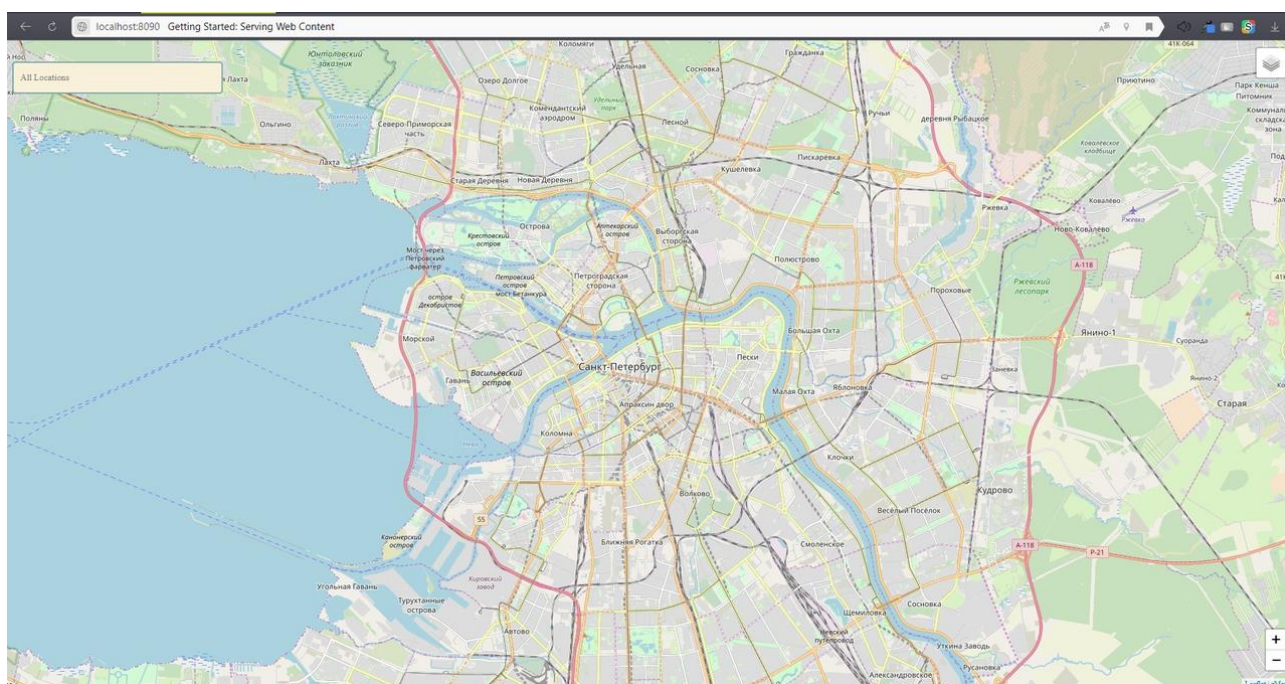


Рисунок – Главная страница сайта

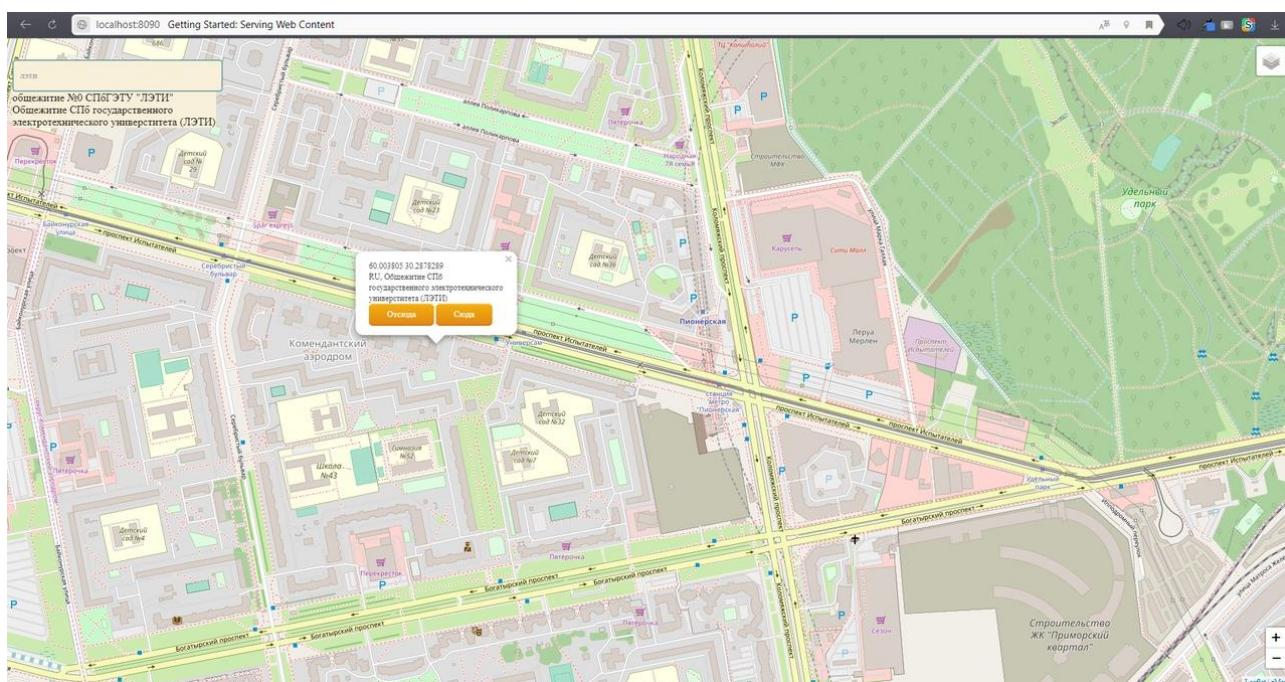


Рисунок – Поиск по названию

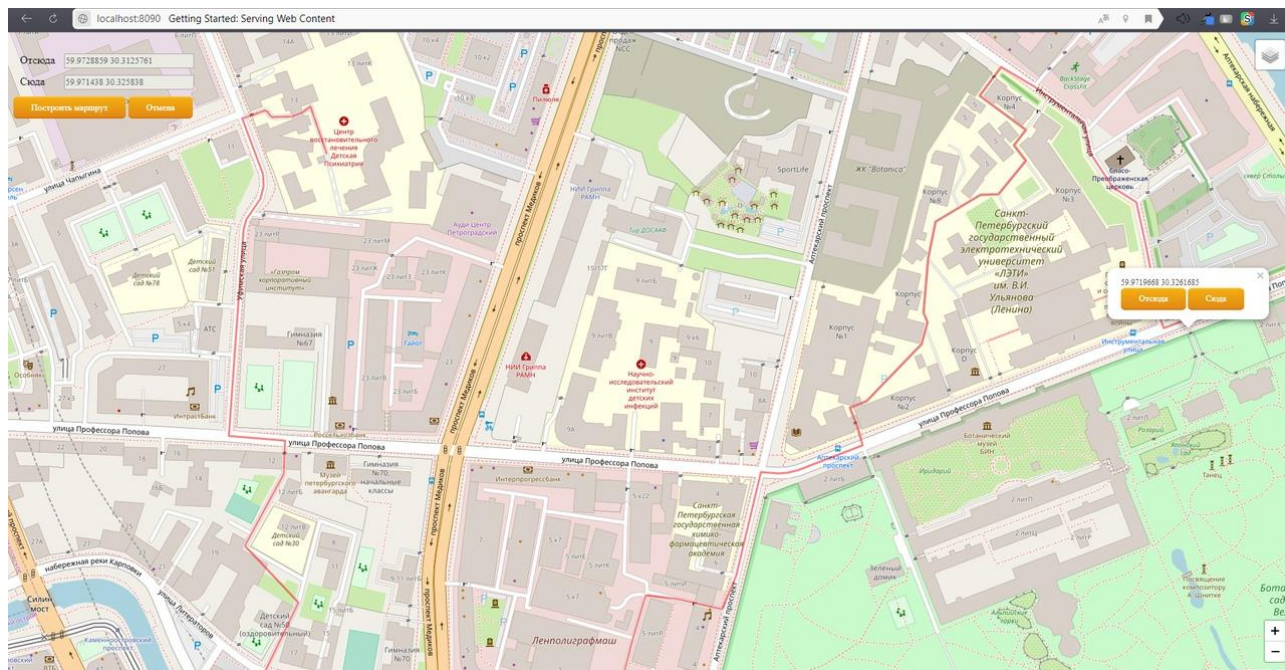


Рисунок – Построение маршрута