

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

УТВЕРЖДАЮ
Проректор по НР
Университета ИТМО

_____ Никифоров В.О.
“ ___ ” _____ 2017 г.

МОДУЛЬ ГОРОДСКИХ ДАННЫХ

РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

ЛИСТ УТВЕРЖДЕНИЯ

Инд. №	
Подп. и дата	
Взам. инв. №	
Инв. № дубл.	
Подп. и дата	

Представители
Организации-разработчика

Руководитель
директор ИДУ

_____ Митягин С.А.
“ ___ ” _____ 2019 г.

Ответственный исполнитель,

_____ **XXXXXXXXXX**

“ ___ ” _____ 2019 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УТВЕРЖДЕН
RU.СНАБ.00023 05-ЛУ

МОДУЛЬ ГОРОДСКИХ ДАННЫХ

Листов __

2019

Инв.№подл.	Подп. и дата	Взам.инв.№	Инв.№дубл.	Подп. и дата

АННОТАЦИЯ

Документ содержит руководство системного программиста программной платформы построения цифровой модели города, предназначенной для автоматизации процессов оценки и анализа городской среды и поддержки принятия решений при планировании изменений.

СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ	4
2.	СТРУКТУРА ПРОГРАММЫ.....	4
3.	НАСТРОЙКА ПРОГРАММЫ.....	5
4.	ПРОВЕРКА ПРОГРАММЫ	6
5.	ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ	6
6.	СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ	6

1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

Программа реализует представление объектов городской среды и их параметров, формирует связи между объектами, а также предоставляет сведения о составе населения проживающего в границах исследуемой территории. Веб интерфейс обеспечивает доступ пользователей к данным, предоставляемым программой за счет реализации отображения объектов городской среды на картографической подложке, а так же предоставляет инструменты для выбора объектов, просмотра их параметров, отображения связей между объектами, просмотра сведений о населении в графическом и табличном виде, просмотра картографических слоев.

2. СТРУКТУРА ПРОГРАММЫ

Программное обеспечение построено по принципу клиент-серверного приложения. Состоит из клиентской части, реализующей пользовательский интерфейс взаимодействия, серверной части, реализующей основные функции обработки данных и базы данных хранящей сведения об объектах. Укрупненная схема взаимодействия компонентов приведены на рисунке 2.1.



Рисунок 2.1 – Логическая структура программного комплекса

Клиентская часть приложения реализована с использованием HTML, Javascript. Для работы с картографической подложкой используются MapBox API и OpenStreetMap. Серверная часть приложения, обеспечивающая обработку запросов и взаимодействие с базой данных реализована на технологиях Java, с использованием фреймворка SpringBoot и ORM фреймворка Hibernate. База данных реализована на PostgreSQL с расширением PostGIS для хранения пространственных данных.

3. НАСТРОЙКА ПРОГРАММЫ

Для корректного функционирования программного комплекса необходимо установить и настроить следующие компоненты:

1. Окружение для исполнения java-программ OpenJDK (версия 1.8);
2. Сервер баз данных PostgreSQL (версия 10.10);
3. Географические расширения для PostgreSQL:
 - 3.1. Поддержка географических объектов - PostGIS (версия 2.4);
 - 3.2. Поддержка геопространственной маршрутизации - pgRouting (версия 2.5);
4. Импортировать дампы базы данных;
5. Собрать приложение с настройками для развертывания;
6. Запустить приложение.

Установка OpenJDK

Для установки JRE в Ubuntu 18.04 выполните:

```
sudo apt install openjdk-8-jre
```

Для проверки установленной версии выполните:

```
java -version
```

Установка PostgreSQL

Для установки PostgreSQL в Ubuntu 18.04 выполните:

```
sudo apt-get install postgresql-10
```

Установка географических расширений

Для установки расширения PostGIS в Ubuntu 18.04 выполните:

```
sudo apt-get install postgis
```

Для установки расширения pgRouting в Ubuntu 18.04 выполните:

```
sudo apt-get install postgresql-10-pgrouting
```

Импортирование базы данных

Для импортирования базы данных проекта после установки PostgreSQL выполните:

```
pg_restore -h localhost -U postgres -F t -C -d postgres <имя файла> , где <имя файла> - имя файла дампа БД.
```

Сборка приложения

Для сборки приложения из исходного кода необходимо выполнить следующие шаги:

- 1) В конфигурационном файле `application.properties` изменить значение настройки `spring.datasource.url` на `jdbc:postgresql://localhost:5432/spb`;
- 2) В конфигурационном файле `application.properties` добавить настройки `hostname=<адрес>`, где `<адрес>` - адрес сервера и `server.port=<порт>`, где `<порт>` - порт приложения;

3) Для сборки проекта в jar-файл выполнить команду:

```
mvn clean package;
```

Запуск приложения

Для запуска приложения необходимо выполнить команду:

```
nohup java -jar <имя файла> > log.txt &, где <имя файла> - имя файла приложения.
```

4. ПРОВЕРКА ПРОГРАММЫ

Проверка программы осуществляется путем вызова программы с использованием интернет-браузера. В адресной строке необходимо указать имя сервера, на котором запущено приложение-сервер. При успешном подключении на экране появится пользовательский интерфейс программы (Рис. 4.1).

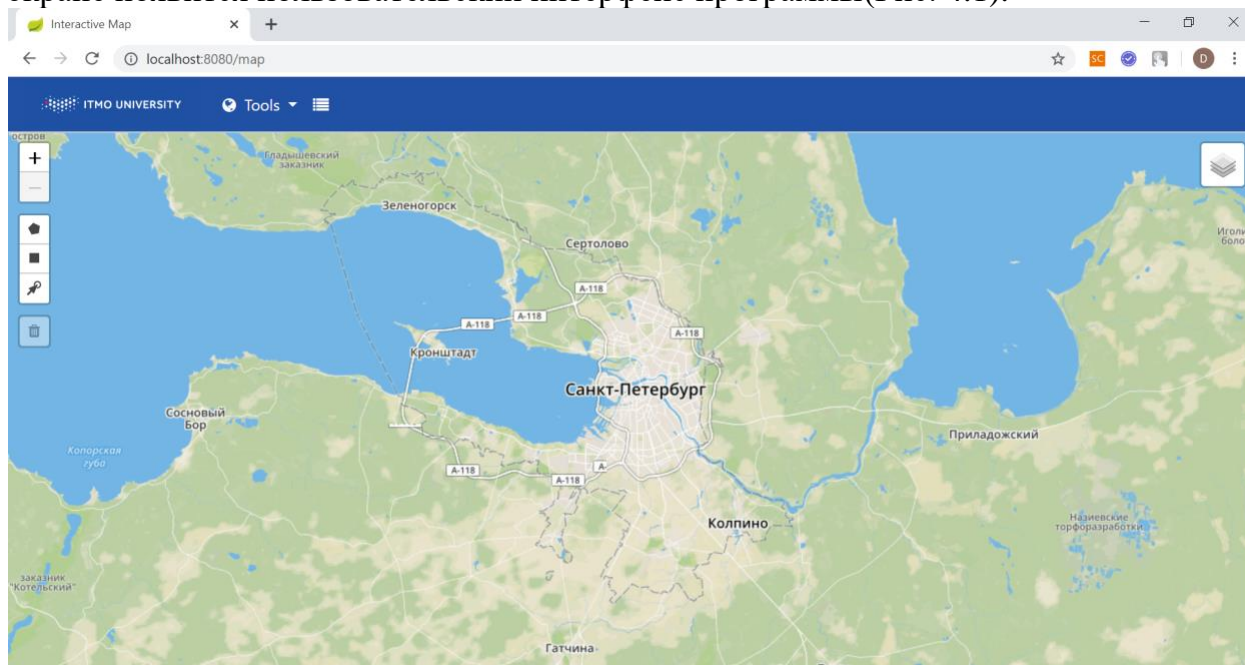


Рисунок 4.1 – Пользовательский интерфейс программы

Если указанные действия привели к появлению на экране формы, аналогичной рис. 4.1, то проверка запуска программы считается пройденной.

5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

Дополнительные возможности отсутствуют.

6. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

Сообщения системному программисту являются стандартными уведомлениями браузера при совершении неверных недопустимых действий (см. рис. 6.1).

Page not found (404)

Request Method: GET

Request URL: http://77.234.213.237:14080/repor1

Using the URLconf defined in project.urls, Django tried these URL patterns, in this order:

1. ^admin/
2. ^points/\$
3. ^points/(\d+)/\$
4. ^points/map/\$ [name='map_frame']
5. ^reports/(\d+)/\$ [name='report']
6. ^reports/(\d+)/image/(\d+)/\$ [name='report_image']
7. ^reports/csv/\$
8. ^tile/\$
9. ^layers/\$
10. ^layers/(\d+)/\$
11. ^objects/(\d+)/\$ [name='objects_report']
12. ^reports/\$ [name='view_reports']
13. ^reports/(\w+)/\$ [name='view_reports']
14. ^app/map/\$ [name='view_map']
15. ^\$ [name='view_index']
16. ^media/(?P<path>.*)\$

The current path, repor1, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

Рисунок 6.1 – Пример сообщения о выполнении некорректных операций на сервере

