Основной текст – основной текст набирается по правилам набора без ошибок. Все предлоги должны быть соединены неразрывным пробелом с относящемуся к ним тексту. Особенно это касается предлогов, располагающихся в конце строки или вначале. В качестве тире используется специальный символ «короткое тире». В тексте применяется автоматический перенос слов. Параметры разметки страницы и остальных основных элементов желательно использовать из настоящего шаблона. Для набора основного текста пояснительной записки предпочтительно использовать стиль «Обычный». Для вставки ссылки на источник литературы используются квадратные скобки из латинского алфавита, например, ссылка на первый источник согласно списку использованных источников [1]. Нумерация в списке использованных источников сквозная. Первому источнику соответствует первый, упомянутый в тексте. Оформление списка – смотри СТП БГТУ 001-2010.

Допускается в тексте использовать перечисления:

* простой список;
* простой список;
* простой список;
* простой список.

Для многоуровневого перечисления сначала используются знаки короткого тире, затем строчные буквы кириллицы, отделяемые от последующего текста круглой скобкой, далее – нумерационное перечисление, также отделяемое круглой скобкой:

* многоуровневый список сначала формируется из простого списка в списке стилей, применительно к данному документу:
  1. многоуровневый список далее вызывается одинарным нажатием клавиши [Tab] на клавиатуре [2]:
     1. далее уровень списка также меняется нажатием клавиши [Tab] на клавиатуре;
     2. и идет по порядку автоматически;
  2. для того, чтобы вернуться на уровень списка, идущего ранее, необходимо сначала перейти на следующий уровень клавишей [Tab], а затем сразу же вернуться нажатием клавиши [Backspace] последовательно два раза.

При оформлении пояснительной записки придерживаются правила единообразия. Если Вы начали использовать квадратные скобки для обозначения, например, клавиш клавиатуры, то и далее в пояснительной записке также применяйте это обозначение во всех разделах. Можно для обозначения клавиш использовать прописные буквы, но одинаково во всей пояснительной записке.

Аналогично и для выделений текста. Не следует использовать разные способы выделения. Если Вы сначала стали использовать курсив, то его в дальнейшем и применяйте в качестве выделения. Не рекомендуется использовать подчеркивание.

Если вводятся какие-то сокращения или условные обозначения, то они должны быть оформлены соответствующим списком условных сокращений или обозначений, и прописаны после первого упоминания в тексте в круглых скобках. Например, пояснительная записка дипломного проекта (ДП) включает титульный лист, задание, реферат (на русском, белорусском и английском языках), содержание, введение и так далее. ДП оформляется в соответствии с требованиями…

Длина концевой строки абзаца основного текста должна быть не менее 25% от длины всей строки. Иначе используйте дополнительные настройки шрифта: Шрифт – Дополнительно –Интервал – Уплотненый / Разреженный – На 0,1–0,3 пт.

Для выравнивания текста по высоте допускается использовать междустрочный интервал в пределах одной страницы в значении Множитель от 0,99 до 0,95 и от 1,01 до 1,05 (пример – см. настройки диалогового окна Абзац для первой страницы текущего документа в контекстном меню к абзацу).

Все рисунки должны иметь порядковую нумерацию в пределах раздела, к которому они относятся и предварительно обозначены в записке. Например, макет сайта приведен на рисунке 1.12. То есть рисунок 1.12 относится к первому разделу пояснительной записки и является двенадцатым по счету. После того, как дана ссылка на рисунок, его рекомендуется вставлять по окончании абзаца. Если размер рисунка не позволяет вставить его сразу после абзаца и его уменьшение до нужных размеров приводит к потере информации, то допускается его перенос на следующую страницу. Но в этом случае свободное поле листа заполняется последующим текстом, а сам рисунок размещается в начале следующей страницы по правилам оформления рисунков. Для самого рисунка рекомендуется использовать стиль Рисунок, а для подрисуночной подписи – стиль Подрисуночная подпись в текущем шаблоне.

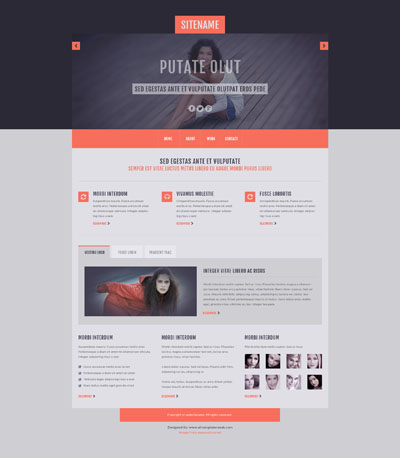


Рисунок 1.12 – Макет сайта компании оранжево-серого цвета

Допускается ссылка на рисунок, размещаемый в приложении. В этом случае само приложение нумеруется прописными буквами кириллицы (например, приложение А), а ссылка на рисунок в приложении соответствует его порядковому номеру в этом приложении. Например, форма регистрации (создание аккаунта) приведена на рисунке А.5. То есть рисунок размещается в приложении А и является пятым по счету в этом приложении.

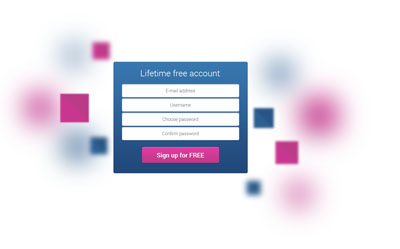


Рисунок А.5 – Форма регистрации (создание аккаунта)

При оформлении таблиц размер шрифта должен быть 14 пт. Допускается поворот таблицы по часовой стрелке на 90° (то есть альбомная ориентация таблицы). В этом случае она должна быть размещена на отдельной странице, следующей после первого упоминания этой таблицы. Оставшийся до следующей страницы интервал заполняется текстом, следующим по смыслу.

Для ссылки на таблицу применяется то же правило, что и для ссылки на рисунок. Например, информация о параметрах используемого графического материала для главной страницы сайта приведена в таблице 4.3.

Таблица 4.3 – Параметры графического материала для сайта

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение файла | Размер изображения, pix×pix | Разрешение, ppi | Расширение файла |
| Logo | 35×35 | 300 | eps |
| Slider | 450×35 | 72 | svg |
| Photo\_№ | 100×100 | 96 | jpg |
| Footer | 1050×35 | 72 | png |

Последующий текст должен быть отделен от таблицы интервалом в 14 пт.

Таблицы, в отличие от рисунков, можно переносить на следующую страницу. В этом случае оформление выглядит следующим образом (таблица 4.4).

Таблица 4.4 – Правила переноса таблицы на следующую страницу

|  |  |  |  |
| --- | --- | --- | --- |
| Заголовок 1 | Заголовок 2 | Заголовок 3 | Заголовок 4 |
| 1 | 2 | 3 | 4 |
| Название 1 | Текст текст текст текст текст текст | Цифры | Цифры |
| Название 2 | Текст текст текст текст текст текст текст текст текст | Цифры | Цифры |

Продолжение таблицы 4.4

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Название 3 | Текст текст текст текст текст текст | Цифры | Цифры |
| Название 4 | Текст текст текст текст текст текст текст текст текст | Цифры | Цифры |

Последняя строка при переносе таблицы не содержит границы снизу! Цифры в таблицах выравниваются по разрядам!!! Для оформления заголовка таблицы и ее продолжения используйте стиль Заголовок таблицы. Чтобы отделить текст от таблицы – стиль Текст после таблицы.

Формулы, при их наличии, набираются в редакторе формул!!! Для этого используем Меню – Вставка – Объект… – MathType 6.0 Equation или Microsoft Equation 3.0. Формулы оформляются в соответствии со стилем Формула. Для того, чтобы она стала строго по центру в начале строки нажимается клавиша [Tab], а после набора формулы еще раз [Tab].

. (4.5)

При наличии обозначений в формуле (стили Обозначение формулы и Продолжение обозначения формулы), которые ранее не указывались в тексте, после формулы ставится запятая и в новом абзаце через слово «где» записывают эти обозначения:

, (4.6)

где *Math* – первое обозначение;

*Soft* – второе обозначение.

Для форматирования разделов используйте стили с соответствующим названием. Стиль Раздел простой используется для форматирования раздела, сразу после которого идет текст (см. раздел 2 – ниже). Если после текста следует подраздел, то для его оформления применяется стиль Подраздел простой. Соответственно, если в подразделе есть пункты и они следуют после текстовой части, то стиль Пункт простой.

Стиль Раздел с подразделом подразумевает, что сразу после раздела стоит подраздел (пример – раздел 1). При этом для оформления самого подраздела используется стиль Подраздел с разделом. Аналогичным образом оформляется подраздел с пунктом (применяются стили – для подраздела: Подраздел с пунктом, а для пункта: Пункт с подразделом).

Для форматирования Реферата, Введения, Содержания, Заключения, Перечня графического иллюстративного материала, Приложений применяется стиль Введение.

Реферат

Пояснительная записка дипломного проекта выполнена в объеме \_\_\_ страниц, \_\_\_ рисунков, \_\_\_\_ таблиц, \_\_\_ использованных источников, \_\_\_ приложений.

Перечень ключевых слов – не менее пяти, до пятнадцати

Объект разработки.

Цель проекта.

Задачи, решаемые в проекте.

Краткие результаты – этапы реализации проекта, область внедрения.

Объем графического материала:

* функциональная схема – 1 лист А3;
* схема базы данных – 1 лист А3;
* конкретно указать наименование плакатов и их формат (допускается А3, А4).

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 00.00.ПЗ

БГТУ

74319ХХ, 2019

Реферат

Н. контр.

Новосельская

у

1

1

Консульт.

Abstract

Explanatory letter of degree project is performed in the content of \_\_\_ pages, \_\_\_\_ figures, \_\_\_\_ tables, \_\_\_\_\_ literary sources, \_\_\_\_ annexes.

Development object is.

The aim of the project

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 00.00.ПЗ

БГТУ

74319ХХ, 2019

Реферат

Н. контр.

Новосельская

у

1

1

Консульт.

Project tasks are.

Main results include .

Content of graphic matter:

* functional diagram – 1 sheet of A3 format;
* database scheme – 1 sheet of A3 format;

Содержание

стр.

[Введение 10](#_Toc9419643)

[1 Аналитический обзор 12](#_Toc9419644)

[1.1 Анализ предметной области 12](#_Toc9419645)

[1.2 Анализ аналогичных решений 12](#_Toc9419646)

[2 Проектирование структуры программного средства 20](#_Toc9419647)

[2.1 Цели задачи программного продукта 20](#_Toc9419648)

[2.2 Проектирование 20](#_Toc9419649)

[3 Разработка программы 27](#_Toc9419650)

[3.1 Разработка back-end части приложения 27](#_Toc9419651)

[3.2 Разработка front-end части приложения 30](#_Toc9419652)

[3.3 Создание инфраструктуры базы данных 33](#_Toc9419653)

[3.4 Выводы по разделу 3 35](#_Toc9419654)

[4 Тестирование 36](#_Toc9419655)

[4.1 Тестирование приложения 36](#_Toc9419656)

[5 Руководство пользователя 41](#_Toc9419657)

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 00.00.ПЗ

БГТУ

74319ХХ, 2019

Содержание

Н. контр.

Новосельская

у

1

1

Консульт.

Введение

В настоящее время всё больше растёт интерес к ГИС — геоинформационным системам, предназначенным для сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации.

Они содержат инструменты, позволяющие пользователям искать, анализировать и редактировать цифровые карты. А также многофункциональные средства анализа сведенных воедино табличных, текстовых и картографических бизнес-данных, демографической, статистической, земельной, адресной и другой информации.

Геоинформационные системы получают все большее распространение в таких областях, как управление природными ресурсами, сельское хозяйство, экология, метеорология, кадастры, городское планирование, землеустройство, экономика, транспорт, оборона.

В качестве систем поддержки принятия решений ГИС помогают улучшить обслуживание клиентов, сохранять высокий уровень конкурентоспособности, повышать прибыльность как коммерческим организациям, чья деятельность зависит от пространственной информации, так и тем, которым анализ геоинформации дает заметные преимущества. ГИС являются эффективным инструментом для выбора мест и определения зон торговли, размещения наружной рекламы и производственных объектов, диспетчеризации и маршрутизации средств доставки, информатизации риэлторской деятельности и т.д.

На сегодняшний день, значительное количество информации состоит из или включает в себя геоданные, то есть различные сведения о распределенных в пространстве или по территории объектах, явлениях и процессах. Работа с такими имеющими координатную привязку характеристиками и является сущностью одной из наиболее бурно развивающихся областей рынка программного компьютерного обеспечения – технологией географических информационных систем.

Целью данного дипломного проекта является создание геоинформационной системы для кафедры лесохозяйствования, которая бы выполняла такие задачи как хранение, загрузка, изменение географической информации Негорельского опытного лесхоза и её представление с помощью наложения на аэрофотографическую карту местности, а также возможность создания маркеров для привязки фотографий местности к конкретным географическим координатам.

Для реализации данной системы используется программное решение Geoserver, база данных PostgreSQL, фреймворк для разработки динамических веб-приложений VueJs, а также библиотеку для работы с картами OpenLayers и библиотеку компонентов VueMaterial. Выбор фремворка для разработки веб-приложения пал в пользу VueJs по причине того, что он позволяет создавать быстродействующие, легко оптимизируемые приложения. Он также предоставляет возможность широкой настройки компонентов и использования синтаксиса jsx, так хорошо известного раз-

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 00.00.ПЗ

БГТУ

74319ХХ, 2019

Введение

Н. контр.

Новосельская

у

1

1

Консульт.

работчикам, практикующим использование React. Также фреймворк обзавёлся довольно большим сообществом разработчиков.

База данных PostgreSQL была выбрана, т.к. она является объектно-реляционной ,поддерживает пользовательские объекты и их поведение, включая типы данных, функции, операции, домены и индексы. Это делает Posgres невероятно гибкой и надежной.

Система разрабатывается с использованием языка программирования JavaScript.

При разработке данного приложения был произведен обзор аналогичных продуктов с целью устранения слабых и добавления сильных сторон приложений схожей тематики в дипломный проект. Структура приложения спроектирована таким образом, чтобы позволить расширять функционал в таких направлениях как, например, добавление новых слоёв, улучшение стилизации и редактирование слоёв посредством веб-интерфейса.

1 Аналитический обзор

1.1 Анализ предметной области

На текущий момент разработано множество различных геоинформационных систем как платных, так и бесплатных. Самые популярные ГИС это настольные приложения, поэтому поиск аналогичных программных средств ориентирован на различные десктопные платформы.

1.2 Анализ аналогичных решений

1.2.1 QGIS

QGIS – свободная кроссплатформенная геоинформационная система, состоящая из настольной и серверной части. В QGIS можно создавать и редактировать векторные данные, а также экспортировать их в разные форматы. Можно просматривать и накладывать друг на друга векторные и растровые данные в различных форматах и проекциях без преобразования во внутренний или общий формат.

В 2007 году QGIS стал официальным проектом Фонда по открытому геопространственному программному обеспечению (OSGeo), миссия которого состоит в том, чтобы содействовать совместной разработке программного обеспечения с открытым исходным кодом для геоматики. Это означало получение командой Quantum GIS организационной поддержки и новых перспектив для развития.

На сегодняшний день QGIS — это зрелый программный продукт, сравнимый с коммерческими аналогами и поддерживаемый международным сообществом разработчиков и пользователей.

К достоинствам данного приложения можно отнести:

* большой выбор инстпументов для создания и редактирования данных;
* поддержка многих языков;
* поддержка большого количества форматов данных;
* поддержка интеграции с многими базами данных;
* поддержка работа с практически всеми системами координат.

К недостаткам можно отнести:

* огромное количество настроек;
* большой размер приложения;
* медленная работа на слабых устройствах.

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 01.00.ПЗ

БГТУ

74319ХХ, 2019

Аналитический обзор

Н. контр.

Новосельская

у

1

15

Консульт.

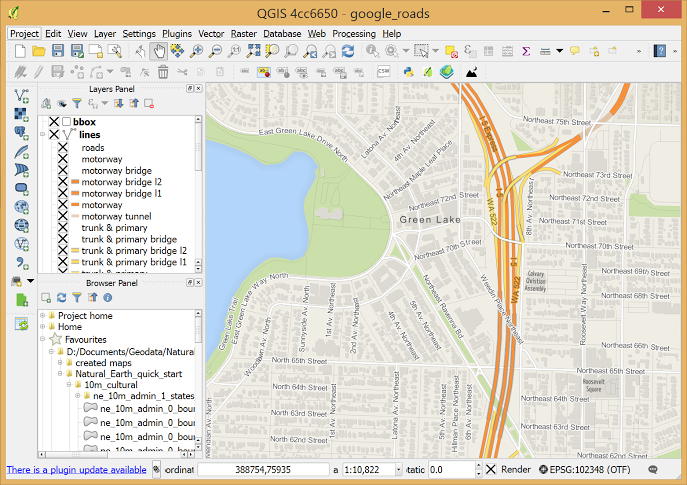


Рисунок 1.1 - Скриншот приложения QGIS

1.2.2 gvSIG

gvSIG — настольная географическая информационная система, предназначенная для сбора, хранения, обработки, анализа и развёртывания любой географически привязанной информации для решения комплексных проблем управления и планирования. gvSIG известна дружественным к пользователю интерфейсом, дающим возможность доступа к наиболее распространенным форматам данных, как векторным, так и растровым. gvSIG имеет обширный набор средств для работы с географической информацией (инструменты запросов, создание макетов, геообработка, сетевой анализ и т.д.)

К положительным качествам данного переводчика можно отнести:

* большой набор инструментов для работы с географической информацией;
* работа с распространёнными форматами данных(векторные и растровые);
* является бесплатной

К отрицательным характеристикам можно отнести следующие:

* отсутствие поддержки некоторых систем координат;
* сложна в обучении.

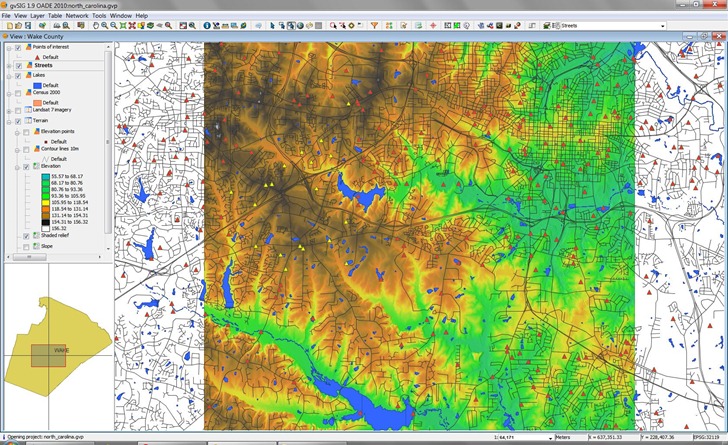


Рисунок 1.2 - Скриншот приложения gvSIG

1.2.3 ArcGIS

ArcGIS – это программное обеспечение для построения ГИС любого уровня. ArcGIS помогает использовать географическую информацию для проведения анализа, лучшего понимания данных и принятия более информированных решений.

Применяется для земельных кадастров, в задачах землеустройства, учёта объектов недвижимости, систем инженерных коммуникаций, геодезии и недропользования и других областях

Основной серверный продукт — ArcGIS for Server, предназначен для многопользовательских геоинформационных проектов с централизованным хранилищем и неограниченным числом рабочих мест, публикации интерактивных карт в Интернете. Для публикации больших объёмов растровых данных выпускается продукт Image Server, для хранения пространственных данных в СУБД и интеграции с другими информационными системами предназначен продукт ArcSDE.

Преимущества приложения:

* возможность создания, редактирования и обмена картами;
* возможность создания и управления базой географических данных;
* cоздание приложений на основании карт.

Недостатки приложения:

* довольно высокая цена;
* большая сложность обучения.

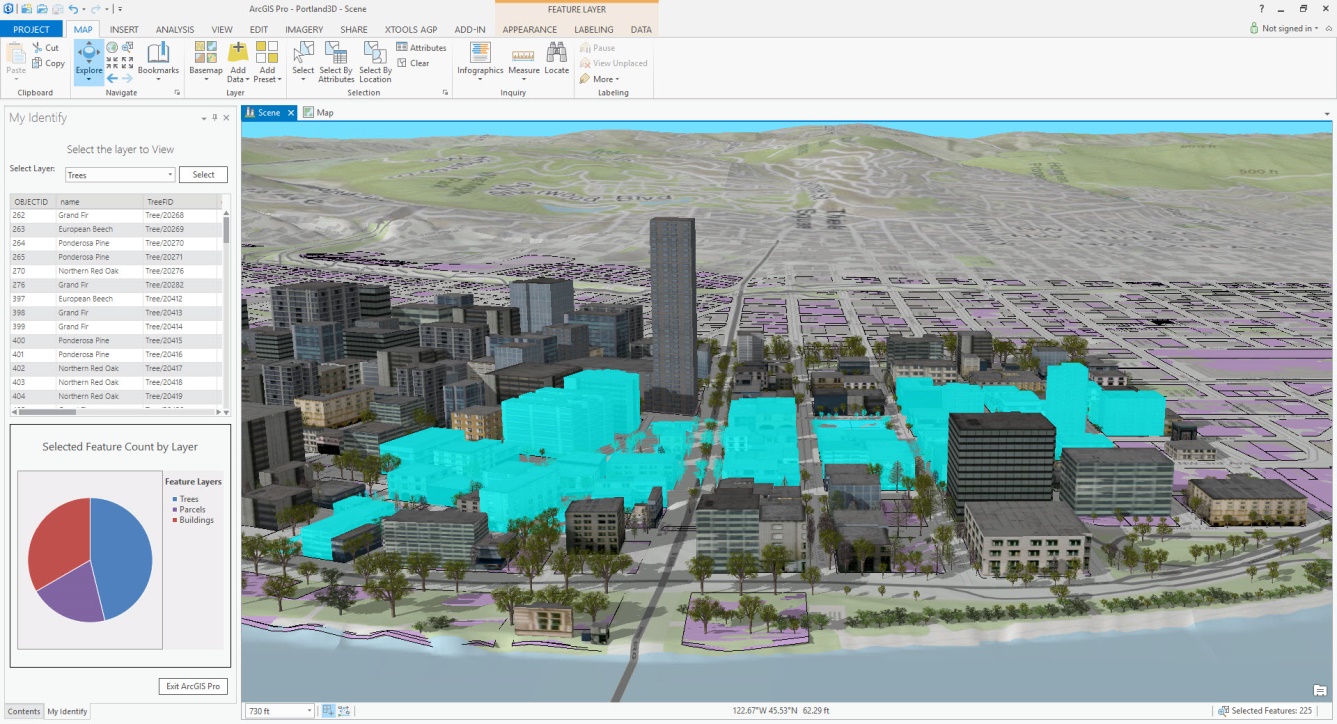


Рисунок 1.3 - Скриншот приложения ArcGIS

1.3 Патентный поиск аналогичных прораммных средств

В соответствии с темой дипломного проекта был проведен патентный поиск в области геоинформационных систем и геоинформационных систем для мониторинга земель. В ходе патентного поиска была изучена научно-техническая и патентная информация:

описание изобретений к патентам;

заявки на изобретения;

рефераты иностранных изобретений.

В результате патентного поиска было найдено два патента. Предметом поиска (объектом исследования или его составной части) была указана “Геоинформационная система”.

Первым найденным патентомв является патент программного комплекса «Геоинформационная система» за номером RU 18586U1 от 13.03.2001, заявителем которого являются Жалковский Евгений Александрович и Страхов Владимир Николаевич. Программный комплекс содержит средства коммуникаций, системное программное и прикладное программное обеспечение, базу метаданных, базы геопространственных данных, средства управления приемом и выполнением заявок потребителей и средства отображения и документирования, отличается тем, что средство управления приемом и выполнением заявок выполнено в виде ядра системы, выходы и входы которого подсоединены к средствам коммуникаций, системному и программному обеспечению, средствам отображения и документирования, а также к базе метаданных и базе геопространственных данных, объединенных в одну подсистему.

Вторым найденным патентом является патент геоинформационной системы мониторинга земель за номером RU 124788 U1 от 06.08.2012, заявителем которого является Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Кубанский государственный университет" (ФГБОУ ВПО "КубГУ") . Программа может быть использована для обработки информации для рационального природопользования, а также в области экологии и применяться при проведении мониторинга состояния плодородия почв.

1.4 Постановка задачи

Целью дипломного проекта является разработка WEB-приложения (интернет сервиса) геоинформационная система Негорельский учебно-опытный лесхоз.

Для достижения цели были поставлены следующие задачи:

* исследовать преимущества и недостатки аналогичных приложений;
* разработать интерфейс приложения;
* разработать прототип приложения на основе интерфейса приложения;
* разработать структуру базы данных;
* разработать архитектуру программной реализации приложения;
* разработать геоинформационную систему Негорельского лесхоза.

1.5Выбор платформы и софта

Основной операционной системой для приложения была выбрана Windows Server 2016.Разработка веб-интерфейса выполняется в среде Web Storm с помощью платформы VueJS. Языком программирования для написания кода является Java script. Для хранения и предоставления данных системы разработана база данных. В данном случае выбрана реляционная база данных PostgreSQL. В качестве ключевого звена приложения, выбрано програмное средство GeoServer.Для связи веб-приложения с базой данных является среда выполнения Go.

1.5.1Среда WebStorm

WebStorm – это одна из самых популярных сред разработки для JavaScript, , разработанная на основе платформы IntelliJ IDEA.

WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами(в том числе, рефакторинг кода JavaScript, находящегося в разных файлах и папках проекта, а также вложенного в HTML). Поддерживается множественная вложенность (когда в документ на HTML вложен скрипт на Javascript, в который вложен другой код HTML, внутри которого вложен Javascript) — то есть в таких конструкциях поддерживается корректный рефакторинг. Данная среда разработки доступна для Windows, OS X и Linux.

1.5.2Платформа VueJS

Vue — это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментамии дополнительными библиотеками.

Важной концепцией Vue являются компоненты. Эта абстракция позволяет собирать большие приложения из маленьких «кусочков». Они представляют собой пригодные к повторному использованию объекты. Если подумать, почти любой интерфейс можно представить как дерево компонентов.

1.5.3Библиотека Vuex

Vuex — библиотека для приложений на Vue.js, которая реализует паттерн управления состоянием Flux. Он служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только предсказуемым образом. Vuex интегрируется с официальным расширением [vue-devtools](https://github.com/vuejs/vue-devtools), предоставляя «из коробки» такие продвинутые возможности, как «машину времени» для отладки и экспорт/импорт слепков состояния данных.

1.5.4Библиотека Vue-Material

Vue Material – библиотека компонентов для VueJS,которая явяется интеграция между спецификациями Vue.js и Material Design! Её можно легко настроить для удовлетворения всех потребностей с помощью простого API.

Преимущества Angular Material:

* компоненты интернационализированы и доступны, чтобы все пользователи могли их использовать;
* простые API, в которых невозможно запутаться;
* поведение хорошо протестировано как с модульными, так и с интеграционными тестами;
* настраивается в пределах спецификации Material Design;
* сведение к минимуму эксплуатационных затрат;
* код является чистым и хорошо документированным, чтобы служить хорошим примером для Angular разработчиков;
* поддержка большинства браузеров.

1.5.5Программная платформа Go

Go – компилируемый многопоточный язык программирования, разработанный внутри компании Google. Разработка Go началась в сентябре 2007 года, его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон, занимавшиеся до этого проектом разработки операционной системы Inferno. Официально язык был представлен в ноябре 2009 года. На данный момент поддержка официального компилятора, разрабатываемого создателями языка, осуществляется для операционных систем FreeBSD, OpenBSD, Linux, macOS, Windows, DragonFly BSD, Plan 9, Solaris, Android. Также Go поддерживается набором компиляторов gcc, существует несколько независимых реализаций. Ведётся разработка второй версии языка.

Язык Go разрабатывался как язык программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам Си и C++. По словам Роба Пайка, «Go был разработан для решения реальных проблем, возникающих при разработке программного обеспечения в Google».

Go создавался в расчёте на то, что программы на нём будут транслироваться в объектный код целевой аппаратной и программной платформы и в дальнейшем исполняться непосредственно, не требуя виртуальной машины, поэтому одним из критериев выбора архитектурных решений была возможность обеспечить быструю компиляцию в эффективный объектный код и отсутствие чрезмерных требований к динамической поддержке.

В результате получился язык, «который не стал прорывом, но тем не менее явился отличным инструментом для разработки крупных программных проектов».

Хотя для Go доступен и интерпретатор, практически в нём нет большой потребности, так как скорость компиляции достаточно высока для обеспечения интерактивной разработки.

1.5.6Библиотека OpenLayers

OpenLayers это библиотека с открытым исходным кодом, написанная на JavaScript, предназначенная для создания карт на основе программного интерфейса (API). Библиотека включает в себя компоненты из JavaScript-библиотек Rico и Prototype JavaScript Framework.

OpenLayers позволяет очень быстро и легко создать web-интерфейс для отображения картографических материалов, представленных в различных форматах и расположенных на различных серверах. Благодаря OpenLayers разработчик имеет возможность создать, к примеру, собственную карту, включающую слои, предоставляемые различными серверами, например, Mapserver, ArcIMS или GeoServer.

Преимущества использования библиотеки:

* карты адаптивны - будут выглядеть везде хорошо, как на десктопах, так и на мобильных устройствах;
* можно совмещать разные типы карт;
* отличная документация;
* простое **API.**

1.6Выводы по разделу 1

В данном разделе проведен аналитический обзор существующих геоинформационных систем, выявлены их достоинства и недостатки, приводятся основные понятия, касающиеся современных геоинформационных систем обработки информации. Патентное исследование позволило выяснить, насколько изучена тема обработки географической информации в данное время.

В соответствии с выполненным анализом сформулированы цель и задачи дипломного проекта.

В результате обзора технических средств, для реализации данного дипломного проекта была выбрана платформа VueJS, выбраны языки программирования JavaScript и Go. При реализации проекта использовались библиотеки Vue Material, OpenLayers которые сочетают в себе эффективность и аккуратность подхода к разработке веб-приложения. Для дизайна пользовательского интерфейса веб-приложения был выбран Vue Material так как он реализует принципы Google Material Design. В качестве серверной части выступает среда выполнения Go. База данных, предназначенная для хранения все нужной информации, предоставляемой веб-приложению используется PostgreSQL.

2 Проектирование структуры программного средства

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 02.00.ПЗ

БГТУ

74319ХХ, 2019

Проектирование интерфейсов программного обеспечения

Н. контр.

Новосельская

у

1

15

Консульт.

2.1 Цели задачи программного продукта

Основной целью данного программного средства является создание  веб-приложения,  которое позволит отображать картографическую и прочую визуальную информацию, относящуюся к Негорельскому лесхозу. Программное средство должно иметь бэк-энд часть, представляющую REST-сервисы к которым будет обращаться фронт-энд часть приложения. Также должен иметься настроенный геосервер и база данных для хранения географических объектов. Интерфейс пользователя  должен соответствовать основным идеям концепции построения внешнего вида приложений  Google Material Design.

Программное средство, разрабатываемое в рамках данного дипломного проекта, в первую очередь, предназначено для создания удобной в использовании геоинфорационной системы Негорельского лесхоза в виде веб-приложения для максимальной совместимости с устройствами. Данное программное средство позволит загружать и просматривать графическую информацию, относящуюся к Негорельскому лесхозу.

Приложение ориентировано на пользователей, которые работают с лесохозяйственной системой негорельского лесхоза, для студентов лесохозяйственного факультета которым нужна информативная визуализация лесных массивов и инструменты для её анализа. Данное программное средство позволит повысить эффективность анализа визуальных данных лесхоза, за счёт стилизации лесных площадей по заданным критериям.

Можно выделить следующие основные требования к программному продукту.

Программное средство должно являть собой веб-приложение, отобржающее Негорельский лесхоз:

* Программный продукт должен позволять отображать и анализировать гео информацию Негорельского лесхоза а также добавлять новую.
* Приложение должно быть расширяемым и простым в поддержке.

2.2 Проектирование

Проектирование программного средства в данном дипломном проекте производилось с помощью UML. UML (англ. Unified Modeling Language — унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML представляет собой графическую нотацию, которая предназначена для

моделирования и описания всех процессов, протекающих в процессе разработки. Основу UML представляют диаграммы, которые различаются по типам и предназначены для моделирования различных аспектов разработки.

Все диаграммы можно условно разделить на поведенческие и структурные. Поведенческие диаграммы отображают процессы, протекающие в моделируемой среде. Структурные диаграммы отображают элементы, из которых состоит система. При этом одни и те же типы диаграмм могут использоваться как для моделирования бизнес-процессов, так и для непосредственного проектирования архитектуры. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью [9]. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода. UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (англ. generalization), агрегация (англ. aggregation) и поведение) и больше сконцентрироваться на проектировании и архитектуре. В UML используются следующие виды диаграмм (для исключения неоднозначности приведены также обозначения на английском языке):

* структурные диаграммы;
* диаграммы поведения;
* диаграммы взаимодействия.

Общая структура компонентов программного средства в данном дипломном проекте представлена в виде компонентов UML (рисунок 2.1).

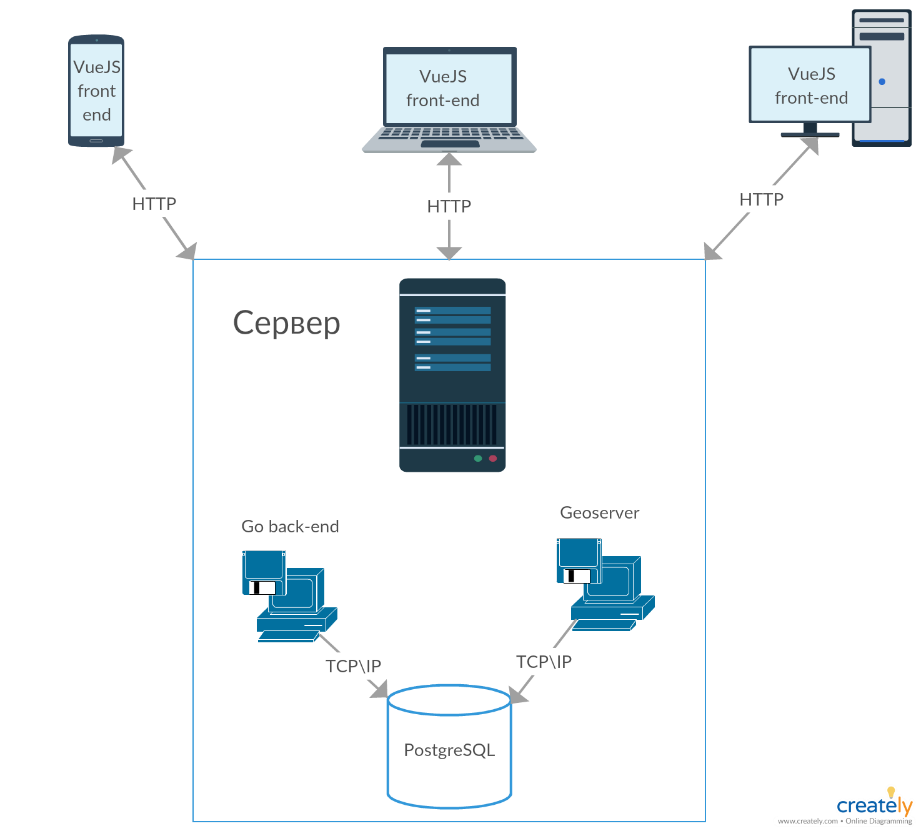


Рис.2.1 Общая структура компонентов приложения

*Front-end* часть представлена веб-приложением, которое созданно для кроссбраузерной работы по отображнию геоданных Негорельского лесхоза. Front-end часть имеет связь с сервисами Back-end части и с REST-сервисами Геосервера.

*Back-end* – это часть системы,представленная REST-сервисами, предоставляющими функциональность авторизации и связующая между базой данных и front-end частью.

Компонент *Geoserver* представляет собой программное средство, предоставляющее для front-end части тайлы, которые преобразуются Геосервером из shape-файлов. Shape-файлы Геосервер получает из базы данных, а отображение производится при помощи front-end части. Также, геосервер предоставляет средства для стилизации тайлов по выбранным признакам, например классам бонитета или возрастом деревьев.

*База данных* – это часть программного средства ,которая предназначена для хранения географической информации. Данная часть должна реализовывать примерно следующий функционал: сохранение информации в формате utf-8, возможность «достать» эту информацию при помощи Геосервера или Back-end части. Также данная часть должна поддерживать большой набор типов объектов и желательно быть объектно-ориентированной.

Общая диаграмма последовательности использования приложения отображена на рисунке 2.2.

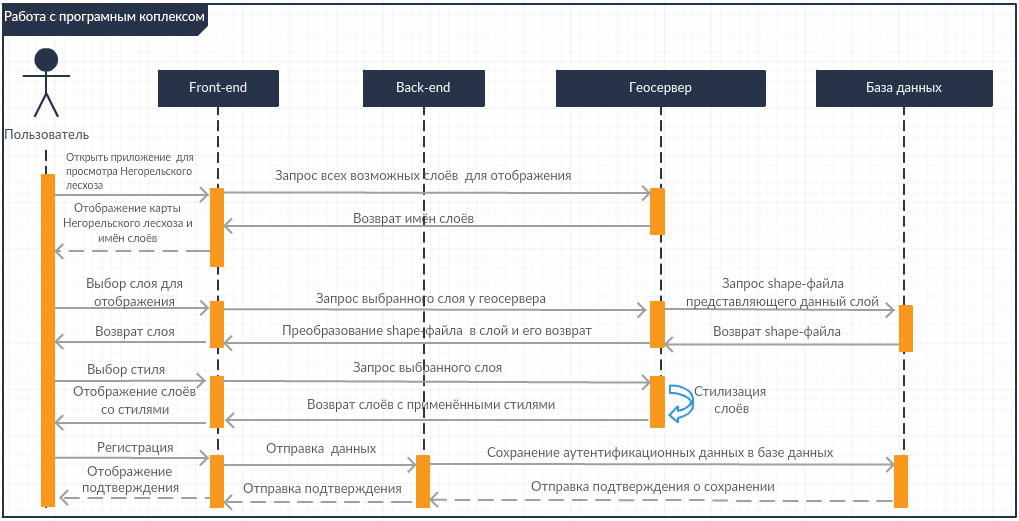


Рисунок 2.2 - Диаграмма последовательности

Последовательность событий начинается с того, что пользователь открывает приложение, при этом отображается карта Негорельского лесхоза, также подгружаются доступные для отображения слои с геосервера. Пользователь может выбрать слой для отображения или выбрать стиль для уже отображаемого. При выборе слоя, происходит запрос к геосерверу, после чего геосервер делает запрос в базу для получения shape-файла, который геосервер преобразовывает в тайлы и присылает в ответе. Данные тайлы отображаются в виде слоя. При отображении слоя, можно выбрать стиль отображения. При выборе стиля, делается запрос к геосерверу, после чего геосервер применяет стиль к выбранным слоям и возвращает стилизованные тайлы.

Ключевыми элементами системы являются Гесервер и back-end части програмного средства,которые язляются так называемым «связующим звеном» между данными, хранящимися в базе данных и пользовательским веб-интерфейсом,который позвляет отобразить данные в удобоваримом для человека виде с целью облегчения анализа.

Для хранения данных используется база данных. Для этих целей была выбрана система управления базами данных PostgreSQL из-за того,что это объектно-ориентированная база данных,которая отлично подходит для хранения данных с любой структурой. Также для PostgreSQL существуют надстройки, позволяющие комфортно работать с shape-файлами и прочей геоинформацией.

Структура таблиц в базе данных зависит от количества и типа полей в shape-файлах,которые мы импортируем в базу.

2.2.1Front-end

Front-end часть приложения спроектирована с применением VueJS , основные параметры приведены в таблице 3.1.

Таблица 3.1 — Описание front-end части приложения

|  |  |
| --- | --- |
| Технология | VueJS |
| Протокол взаимодействия | HTTP |
| Назначение | Отображение карт, слоёв и прочей информации на стороне клиента |
| Дополнительные библиотеки | VueMaterial, Vuex, OpenLayers |

2.2.2Back-end

Back-end часть приложения спроектирована с применением Go , основные параметры приведены в таблице 3.2.

Таблица 3.2 — Описание back-end части приложения

|  |  |
| --- | --- |
| Технология | Go |
| Протокол взаимодействия | HTTP |
| Назначение | Связь клиентского приложения с базой данных, работа в качестве сервера статических файлов, в том числе и клиентского приложения |
| Дополнительные библиотеки | Chi, PQ |

2.2.3База данных

В качестве базы данных используется PostgreSQL. Основные параметры приведены в таблице 3.3.

Таблица 3.1 — Описание базы данных приложения

|  |  |
| --- | --- |
| Технология | PostgreSQL |
| Протокол взаимодействия | TCP\IP |
| Назначение | Хранение пользовательской информации и географической информации приложения |
| Дополнительные библиотеки | PostGIS |

База данных имеет две постоянные таблицы, так как остальные формируются при импорте shape-файлов. На рисунке 3.1 представлена схема таблиц базы данных.

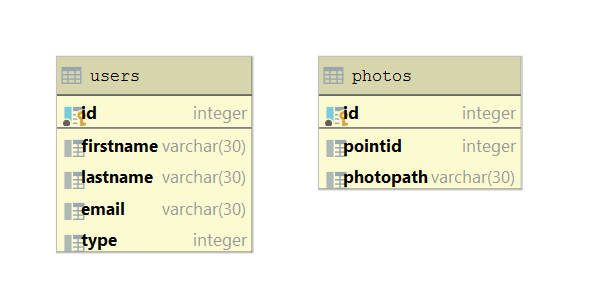


Рисунок 2.2 Схема постоянных таблиц базы даных

Таблица «users» предназначена для хранения пользователей. Описание полей «users» приведено в таблице 2.5

Таблица 3.1 — Описание полей таблицы «users»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | Integer | Первичный ключ |
| Firstname | Varchar(30) | Имя пользователя |
| Lastname | Varchar(30) | Фамилия пользователя |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| Email | Varchar(30) | Электронная почта пользователя |
| Type | integer | Тип пользователя |

. Таблица «photos» предназначена для связи точек, отмеченных на карте Негорельского опытно-учебного лесхоза и фотографий этих мест. Описание полей «photos» приведено в таблице 2.6

Таблица 3.1 — Описание полей таблицы «Photos»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Id | Integer | Первичный ключ |
| PointId | Integer | Id точки |
| PhotoPath | Varchar(30) | Фамилия пользователя |

2.2.4Geoserver

В качестве центрального компонента приложения используется программное средство Geoserver. Основные параметры приведены в таблице 3.4.

Таблица 3.1 — Описание базы данных приложения

|  |  |
| --- | --- |
| Технология | Geosever |
| Протокол взаимодействия | HTTP |
| Назначение | Обработка геоинформации для предоставления клиентской стороне |
| Дополнительные библиотеки | Плагины для расширения функциональности geosever |

2.3 Выводы по разделу 2

Исходя из требований разрабатываемого программного средства, можно сделать вывод о том, что хранение данных в приложении лучше всего организовать посредством СУБД PostgreSQL. Данная система организовывает хранение данных в таком виде, который позволит максимально быстро реализовывать вставку, обработку и извлечение данных. Проектирование программного средства производилось в программе Sparx Enterprise Architect с помощью UML-диаграмм. Это позволило составить архитектуру приложения максимально приближенно к реализации программного средства.

3 Разработка программы

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 02.00.ПЗ

БГТУ

74319ХХ, 2019

Проектирование интерфейсов программного обеспечения

Н. контр.

Новосельская

у

1

15

Консульт.

3.1 Разработка back-end части приложения

Для реализации back-end части был выбран Go. Одной из ключевых возможностей языка Go является возможность работы с сетевыми сервисами: отправлять запросы к ресурсам в сети и, наоборот, обрабатывать входящие запросы. Основной функционал по работе с сетью представлен пакетом net. Этот пакет предоставляет различные низкоуровневые сетевые примитивы, через которые идет взамодействие по сети.

Для упрощения маршрутизации был выбран Сhi. Сhi - это легкий, идиоматичный и составной маршрутизатор для создания сервисов Go HTTP. Это особенно полезно для того, чтобы написать большие сервисы API REST, которые поддерживаются в обслуживании по мере роста и изменения проекта. Сhi построен на новом пакете контекста, представленном в Go 1.7, для обработки значений сигнализации, отмены и области запроса в цепочке обработчиков.

Целью создания Chi было найти элегантный и удобный дизайн для написания REST API-серверов, написанный во время разработки службы Pressly API, которая поддерживает нашу общедоступную службу API, которая, в свою очередь, поддерживает все клиентские приложения.

Структура приложения представлена на рисунке 3.1

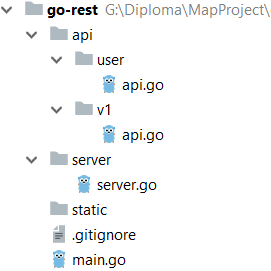


Рисунок 3.1 Структура back-end части приложения

Первоначально запускаемым файлом является «main.go», это так называемая стартовая точка приложения. Первоначально для обработки входящих запросов создаётся обработчик маршрутов роутера при помощи вызова NewRoter из пакета

«server.go» и вызывается функция ListenAndServe для прослушки на указанном порту.

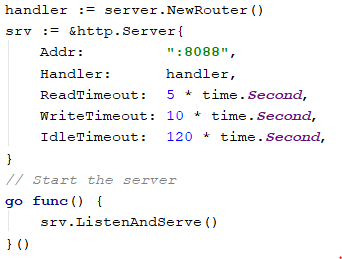


Рисунок 3.2 – Создание сервера

При вызове NewRouter из пакета server, происходит создание обработчиков маршрутов. При указании в url запроса «/user», происходит вызов обработчиков, которые относятся к пакету «user». Также настраивается таймаут и создаётся обработчик, который обращается к серверу для раздачи статических файлов, по этому пути будет раздаваться front-end часть приложения.

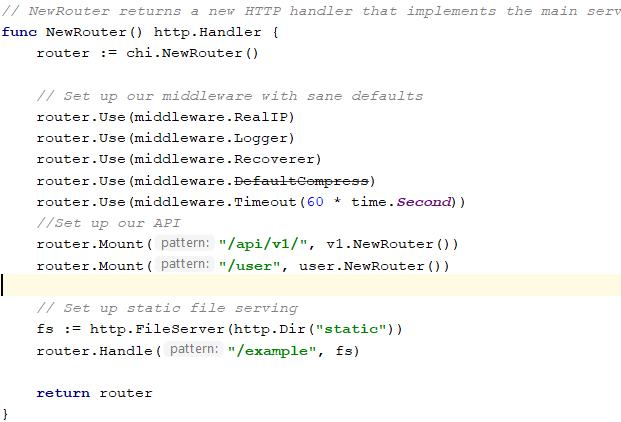


Рисунок 3.3 – Создание роутера и привязка обработчиков

Для реализации работы с базой данных в данном дипломном проекте был выбран пакет PQ. Пакет pq - это драйвер Go Postgres для пакета database / sql.

Подобно libpq, при установлении соединения с использованием pq нужно указать строку соединения, содержащую ноль или более параметров. Подмножество параметров соединения, поддерживаемых libpq, также поддерживаются pq. Кроме того, pq также позволяет указывать параметры времени выполнения.

Для работы с базой данных с помощью pq необходимо выполнить следующие этапы.

Создать строку подключения ,указав необходимы параметры.

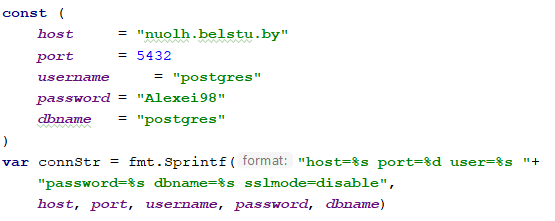


Рисунок 3.4 – Создание строки подключения

Далее создаётся подключение к базе данных и оно остаётся активным до отключения.

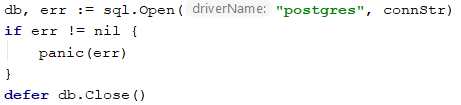


Рисунок 3.5 – Создание подключения

Данный объект подключения в дальнейшем используется для работы с базой данных. Например, для выполнения sql запроса используется метод db.Query.

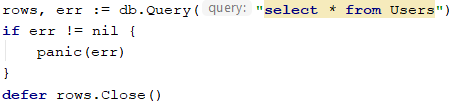


Рисунок 3.6 – Вызов sql конструкции

Основные компонеты приведены в таблице 3.4.

Таблица 3.1 — Описание компонентов back-end части приложения

|  |  |
| --- | --- |
| Компонент | Функция |
| Main.go | Стартовая точка приложения |
| Server.go | Роутер для разделения обработки логики согласно маршруту |
| User/api.go | Пакет с функциями обработки авторизации и регистрации |
| V1/Api.go | Пакет с остальными функциями |

Таким образом обеспечивается работа back-end части приложения и её взаимодействие с базой данных.

3.2 Разработка front-end части приложения

Front-end часть была написана на JavaScript. Для ускорения разработки был выбран фреймворк VueJS. Это фреймворк для создания пользовательских интерфейсов. В отличие от других монолитных фреймворков, Vue спроектирован с нуля для постепенного принятия.

Основным элементом этой части является система отображения карт и слоёв. Система использует библиотеку OpenLayers для работы с тайлам и удобного представления информации в графическом виде.

Основные структурные компоненты приведены в таблице 3.2.

Таблица 3.2 — Описание компонентов front-end части приложения

|  |  |
| --- | --- |
| Компонент | Функция |
| Папка node\_modules | Папка с npm модулями |
| Папка public | Папка с html файлом, к которому подключаются бандлы и остальная метаинформация |
| Папка app | Папка с корневым компонентом приложения |
| Папка assets | Папка с картинками |
| Папка components | Папка с компонентами |
| Папка views | Папка с компонентами-страницами |
| Файл main.js | Инициализационный файл приложения |
| Файл router.js | Файл роутера |
| Файл store.js | Файл хранилища Vuex |

Структура веб-приложения представлена на рисунке 3.1

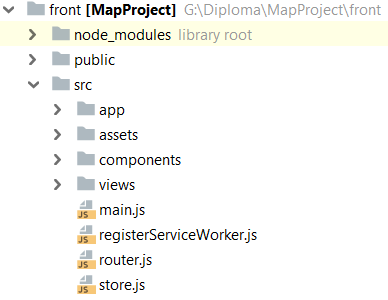


Рисунок 3.7 Структура front-end части приложения

Папка node\_modules хранит в себе необхоимые модули для приложения,один из которых это OpenLayers. В папке public сосредоточены такие файлы как главный HTML файл,который выводится при открытии приложения,фал манифеста и другие файлы с метаинформацией. Папка app содержит корневой компонент приложения,в который так или иначе подключаются другие компоненты. Папка assets содержит и прочие общие ассеты. В папке components находятся компоненты – маленькие элементы так называемого «конструктора», из которых позже собираются страницы,которые находятся в папке views. Файл main.js содержит код инициализации приложения,которое принимает в себя список маршрутов, описанный в файле router.js и файл VUEX-хранилища, находящегося в файле store.js.

Остальные файлы являются конфигурационными и предназаначены для разработки и развертывания, а также для написания правил проверки валидности и семантики кода.

Реализация отображения карты и слоёв описана в компоненте map, логика которого находится в фале map.js.

Первоначально происходит запрос всех доступных для отображения слоёв с геосервера, после чего формируются структуры данных, которые представляют эти слои на клиенте.

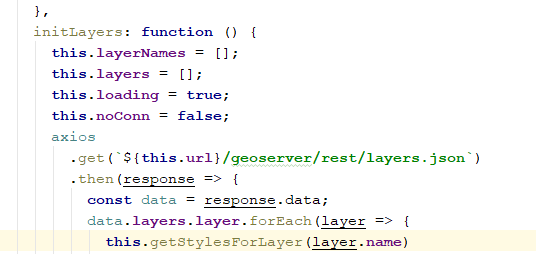


Рисунок 3.8 – Запрос доступный слоёв

После запроса доступных слоёв, необходимо получить стили для каждого слоя, которые могут быть применены. Для этого при получении слоёв идут запросы по именам слоёв к геосерверу, после чего получаем список слоёв и привязанных к ним стилей.



Рисунок 3.9 – Запрос доступных стилей

После получения стилей для каждого слоя, формируется таблица, при помощи которой можно выбирать отображаемые слои и применяемые к ним стили. Также, предварительно загружаются две карты местности, схематичная и аэрофотографическая, которые для удобства занимают половину экрана. После выполнения вышеперечисленных действий, можно выбирать или скрывать слои, стилизовать их, поворачивать, масштабировать или передвигать карту и прочим образом взаимодействовать с программой.

3.3 Создание инфраструктуры базы данных

В разрабатываемом программном средстве данные хранятся в базе данных. Так как основные данные это географическая информация из shape-файлов, то необходима поддержка базой данных географических типов данных, таких как линия, точка, полигон и так далее. Для внедрения поддержки таких типов данных, существует плагин PostGIS.

Первоначально, после установки базы, необходимо установить PostGIS плагин, его можно поставить, скачав с официального сайта, либо через утилиту Stack Builder. На официальном сайте также можно найти информацию относительно того, какая версия PostGIS будет работать с конкретной версией PostgreSQL. Далее при необходимости создаём базу данных.



Рисунок 3.10 – Создание базы данных

Далее необходимо проинициализировать PostGIS.



Рисунок 3.11 – Инициализация PostGIS

У нас в базе данных появится таблица spatial\_ref\_sys, в которой сразу будет почти 4 тысячи записей. Каждая запись соответствует некоторой пространственной системе координат, которая определяет проекцию долготы и широты на плоскость.

С помощью этой таблицы PostGIS может конвертировать данные из одной проекции в другую.

Далее появляется возможность загружать shape-файлы в базу данных. Проще всего это сделать при помощи утилиты, которая поставляется вместе с PostGIS.

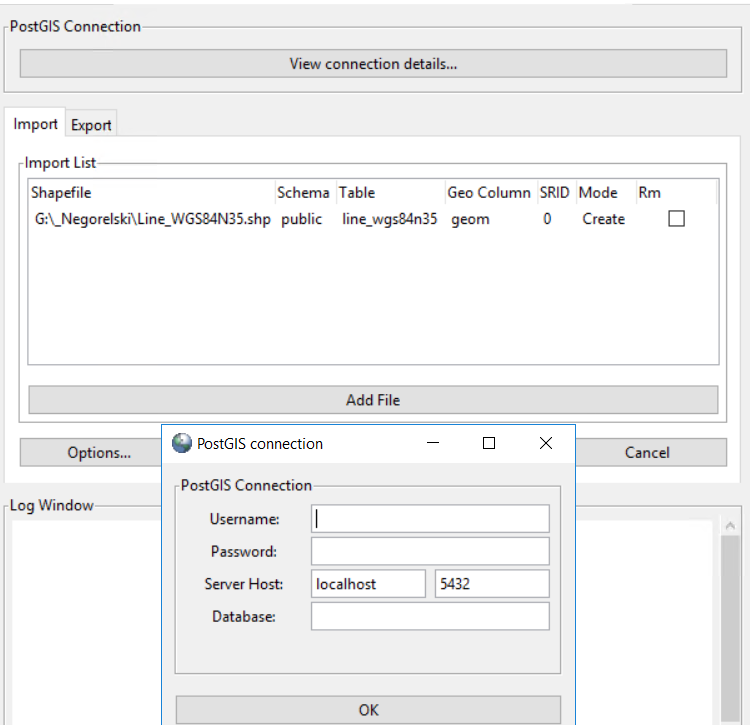


Рисунок 3.12 – Импорт shape-файла в базу данных

Для импорта файла необходимо настроить соединение с базой данных, указав имя пользователя, имя базы, имя хоста и порт соединения. Также необходимо выбрать файл для импорта и поменять поле SRID на необходимое для конкретной системы координат.

После импорта файла в базу данных, появляется извещение о успешности.

Данный представляются в базе данных в виде таблиц, где каждый столбец в таблице является отражением полей в shape-файле. Данные представленные в базе данных, позже используются Geoserver.

3.4 Выводы по разделу 3

В данном разделе создана инфраструктура для работы с геометрическими типами данных.

Разработана программа для работы с геоданными Негорельского лесхоза, которое может выполнять широкий спектр задач по отображению, редкатированию и анализу географических.

В процессе разработки была рассмотрена и настроена организация взаимодействия геосервера, базы данных, веб-интерфейса и сервисов, в конечном итоге образовавших новую геоинформационную систему Негорельского опытно-учебного лесхоза.

4 Тестирование

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 02.00.ПЗ

БГТУ

74319ХХ, 2019

Тестирование

Н. контр.

Новосельская

у

1

15

Консульт.

4.1 Тестирование приложения

На данный момент веб завоевал огромную нишу на рынке приложений. Количество веб-приложений давно превышает количество настольных или мобильных.

Пользователь ожидает, что используемые им приложения просты, интуитивно понятны и работают без сбоев. Поэтому качество приложения играет очень большую роль в его популярности.

Качество приложения – это совокупность характеристик данного приложения, относящихся к его способности удовлетворять установленные и предполагаемые потребности пользователей. Оценить уровень качества приложения и выявить различные ошибки в его работе можно с помощью тестирования.

Тестирование – это поиск дефектов или багов в приложении. Термин «баг» используют для обозначения ошибки, из-за которой приложение выдает неожиданное поведение.

Планируя процесс тестирования, тестировщик часто составляет тест-кейсы. В самом простом случае тест-кейс включает в себя:

краткое описание (то, что тестируем);

* последовательность действий (шаги, которые выполняет тестировщик,чтобы проверить какую-либо функциональность);
* ожидаемый результат (то, что должно получиться после выполнения действий).

В дальнейшем по данным тест-кейсам тестируется приложение. Выполняя последовательность действий из тест-кейса, тестировщик получает фактический результат, который сравнивает с ожидаемым. Если данные результаты не совпадают, то это означает, что тестировщик нашел баг.

Для тестирования был выбран тест-кейс план, представленный в таблице 4.1.

Таблица 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| Номер тест-кейса | Описание тест-кейса | Ожидаемый результат | Полученный результат |
| 1 | Зайти в приложение и выбрать слой для отображения | Должен отобразиться выбранный слой поверх карты. | Приложение открылось успешно, после выбора слоя для отображения, слой отобразился поверх карты |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Выбрать для отображаемого слоя другой стиль. | Стилизация выбранного слоя должна поменяться | При выборе другого стиля для слоя, поменялась стилизация |
| 3 | Выбрать файл для загрузки в базу данных через утилиту PostGis | Файл должен загрузиться в базу данных в кодировке UTF-8 и стать доступным для публикации на геосервере | При загрузке файла в базу данных через утилиту, он появляется в базе данных в кодировке UTF-8 и появляется в числе доступных для публикации слоёв на геосервере |
| 4 | Опубликовать файл на геосервере | Файл должен опубликоваться на геосервере, должен появиться слой, доступный для просмотра, этот слой должен стать доступен в веб-приложении для отображения | Файл опубликовался на геосервере, появился слой, который стал доступен для выбора в качестве отображаемого в веб-приложении |
| 5 | Подключиться к базе данных при помощи приложения QGIS и изменить один из слоёв, который опубликован на геосервере. | Слой должен автоматически обновиться и загружаться в веб-приложении будет уже новый | Слой обновлённый при помощи приложения QGIS в базе данных, обновился и в веб-приложении |
| 6 | Привязать стиль к опубликованному слою | Новый стиль должен появиться в списке выбора стилей для отображаемых на карте | После привязки нового стиля к опубликованному слою, этот стиль стал доступен для выбора в списке стилей для отображаемого слоя |
| 7 | Выбрать слой с выделами и после отображения данного слоя выбрать один из выделов | После щелчка по выбранному выделу, должен осуществиться переход к указанному выделу в приложении Дреко А. | После выбора выдела и щелчка по нему, произошёл переход к выбранному выделу в приложение Дреко А. |
| 8 | Выбрать точечный слой из списка и после отображения, щёлкнуть по одной из точек | После щелчка по выбранной точке, должен загрузиться список фотографий принадлежащих к этому месту | После выбора точки, загрузился список фотографий, принадлежащих к этому месту |
| 9 | Переместить ползунок в верхней части экрана, на котором отображается карта | После смещения ползунка, должны поменяться пропорции подложек, одна из них должна занимать большую часть подложки | После смещения ползунка, одна из карт стала занимать на подложке больше места |
| 10 | Попытаться загрузить слои при отключенном интернете | Должно появиться оповещение о том, что невозможно связаться с геосервером и предложение попробовать заново, при подключении к интернету, слои должны загрузиться | При отключении интернета, появляется оповещение о невозможности подключения к геосерверу и предложение о рестарте загрузки, после возобновления соединения, слои загружаются |

4.2 Вывод по разделу 4

Для тестирования был выбран метод прохождения тест-кейсов, так как это наиболее понятный, визуализировнный и удобный способ.

Тест-кейс – это такое описание проверки работы системы, которое может выполнить любой человек из команды, будь то тестировщик, разработчик, аналитик или даже бизнес-заказчик.

Набор тест-кейсов называется тестовым набором (test suite). Иногда этот набор некорректно называют тест-планом. Тест-план — это именно план: когда, что, зачем, какими ресурсами

Стандартные атрибуты тест-кейса :

* номер – уникальный идентификатор тест-кейса. Его удобно использовать для одинакового понимания, о какой проверке идет речь (например, дать ссылку в баге);
* описание – описание действий, которые необходимо выполнить, но прямого отношения к проверке они не имеют (например, зарегистрироваться в системе для проверки создания элемента). Если предварительных шагов нет, то секция не заполняется;
* ожидаемый результат – описание действий, необходимых для проверки (например, создание элемента);
* Полученный результат – сама проверка: что мы ожидаем получить после выполнения шагов ("Элемент создан").

В результате пройденных тест-кейсов можно убедится, что основной функционал выполняется и все функции работают.

5 Руководство пользователя

Листов

Изм

Лист

№ документа

Иванов

Брусенцова

Романенко

Подп.

Дата

Разраб.

Пров.

Утв.

Лит.

Лист

ДП 02.00.ПЗ

БГТУ

74319ХХ, 2019

Руководство пользователя

Н. контр.

Новосельская

у

1

15

Консульт.

5.1 Руководство администратора системы

В данной программе предусмотрены пользователи двух видов:

* рядовой пользователь
* пользователь-администратор

Рядовой пользователь – это человек, который имеет право только просматривать информацию в данной системе. Администратор обладает большим набором привилегий, таких как:

* просмотр информации как обычный пользователь
* добавление информации
* модификация существующей информации
* удаление информации

Для администрирования системы можно пользоваться несколькими частями программы. Самым мощным инструментом для администрирования данной системы является Geoserver.

Первоначально, необходимо открыть веб-панель мониторинга и настройки Geoserver. При открытии необходимо ввести аутентификационные данные. Страница показана на рисунке 5.1

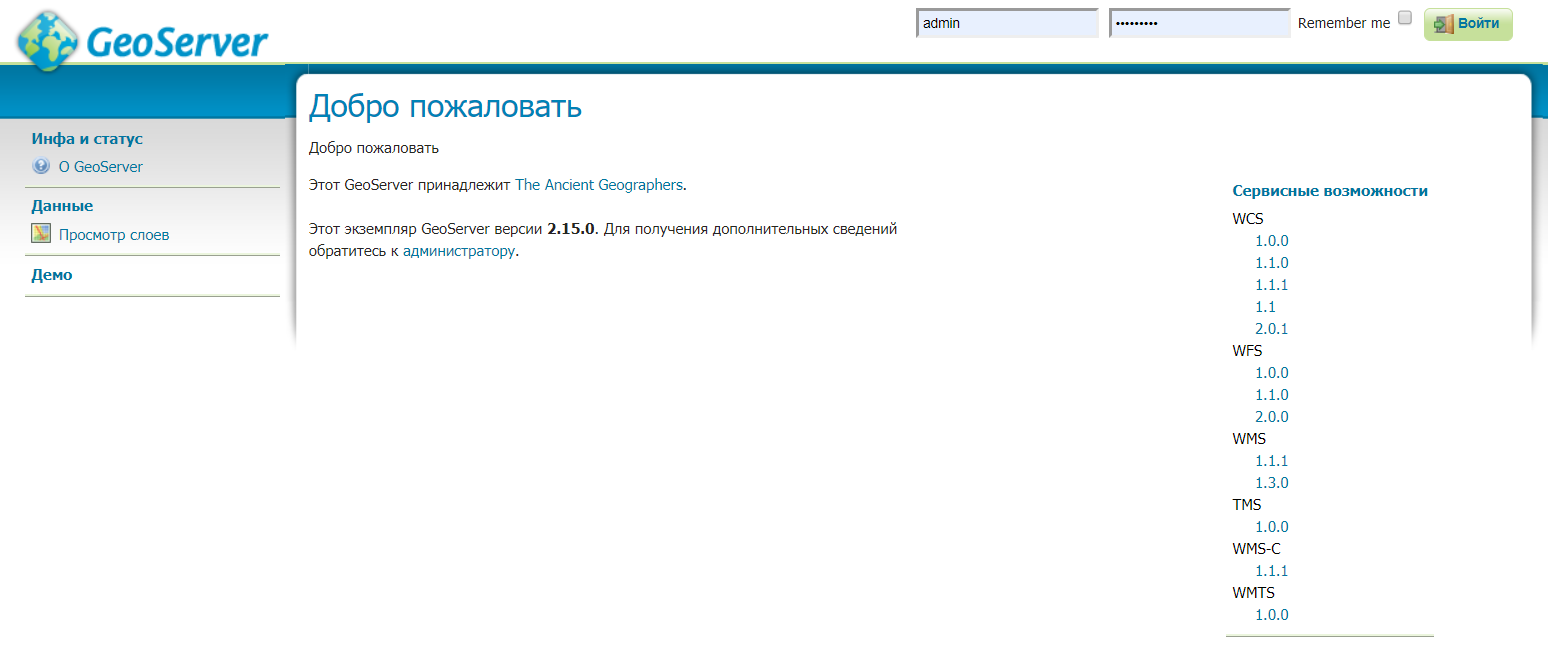


Рисунок 5.1 – Открытие стартовой страницы Geoserver

После ввода аутентификационных данных совершается переход на главную страницу администрирования Geoserver. Эта страница представлена на рисунке 5.2

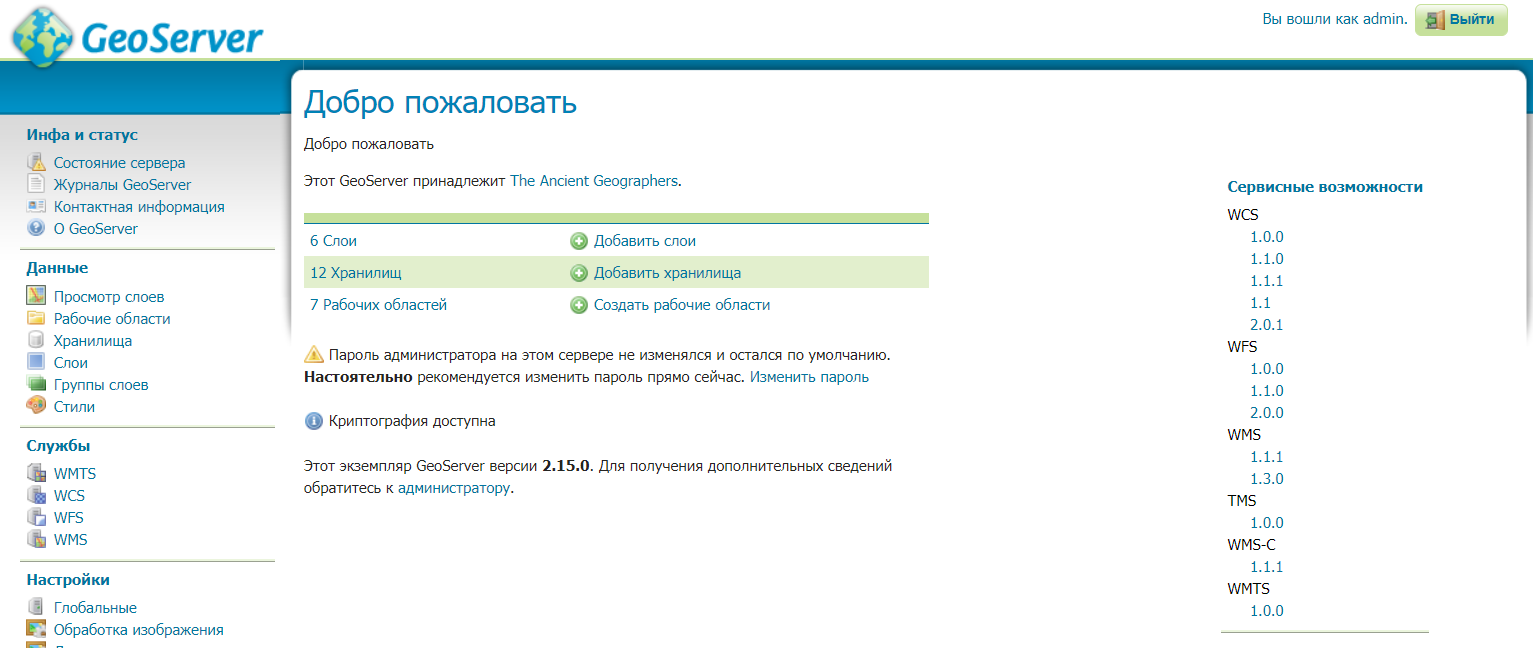


Рисунок 5.2 Главная страница администрирования Geoserver

С данной страницы, администратор может перейти к добавлению источников данных, публикации слоёв, изменению тематических стилей и так далее.

Допустим, администратор хочет добавить новый источник данных. Для этого необходимо перейти в «хранилища» (рисунок 5.3)

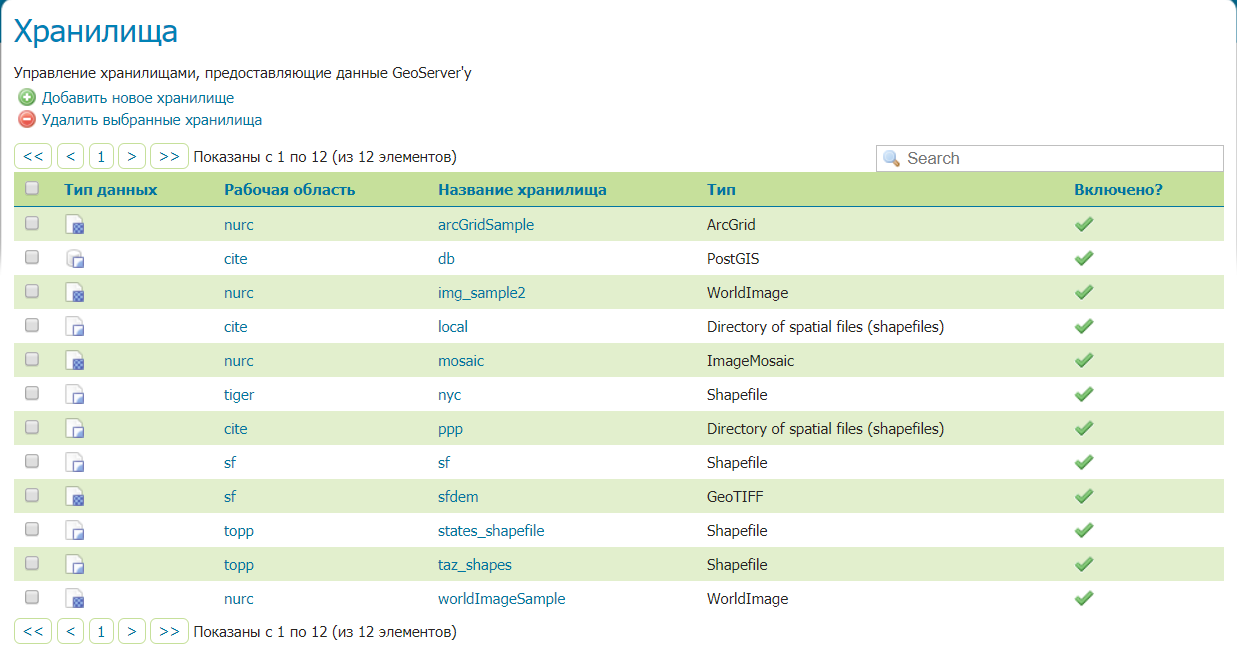


Рисунок 5.3 Хранилища

«Хранилища» представляют собой страницу с отображёнными на ней существующими источниками данных. В таблице указывается тип хранилища, его название и его рабочая область. Существующие хранилища можно отключить от геосервера, просто выбрав чекбокс напротив нужного и нажав на кнопку «Удалить выбранные хранилища». Для добавления нового хранилища, необходимо просто нажать на кнопку «Добавить новое хранилище», после чего происходит переход к форме добавления выбора типа нового хранилища, показанной на рисунке 5.4

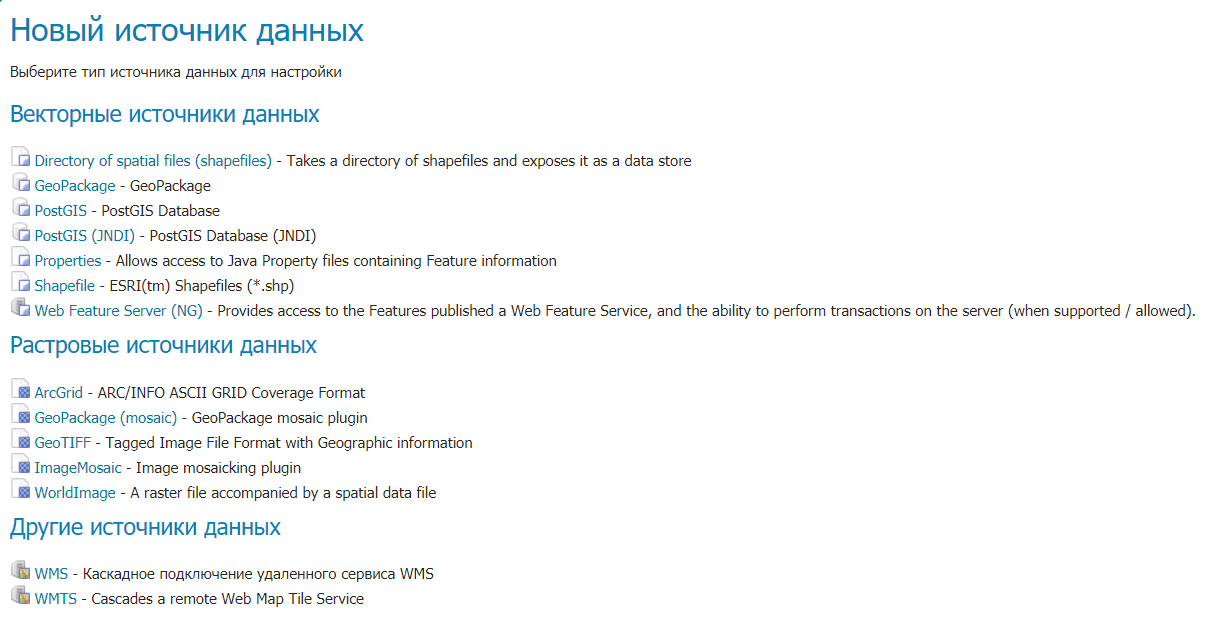


Рисунок 5.4 Выбор типа источника данных

Здесь можно выбрать как одиночные shape-файлы, директории с shape-файлами, так и уже существующие сервера, отдающие shape-файлы, а также базы данных и прочее. После выбора типа источника данных, происходит переход непосредственно к форме добавления источника, показанный на рисунке 5.5

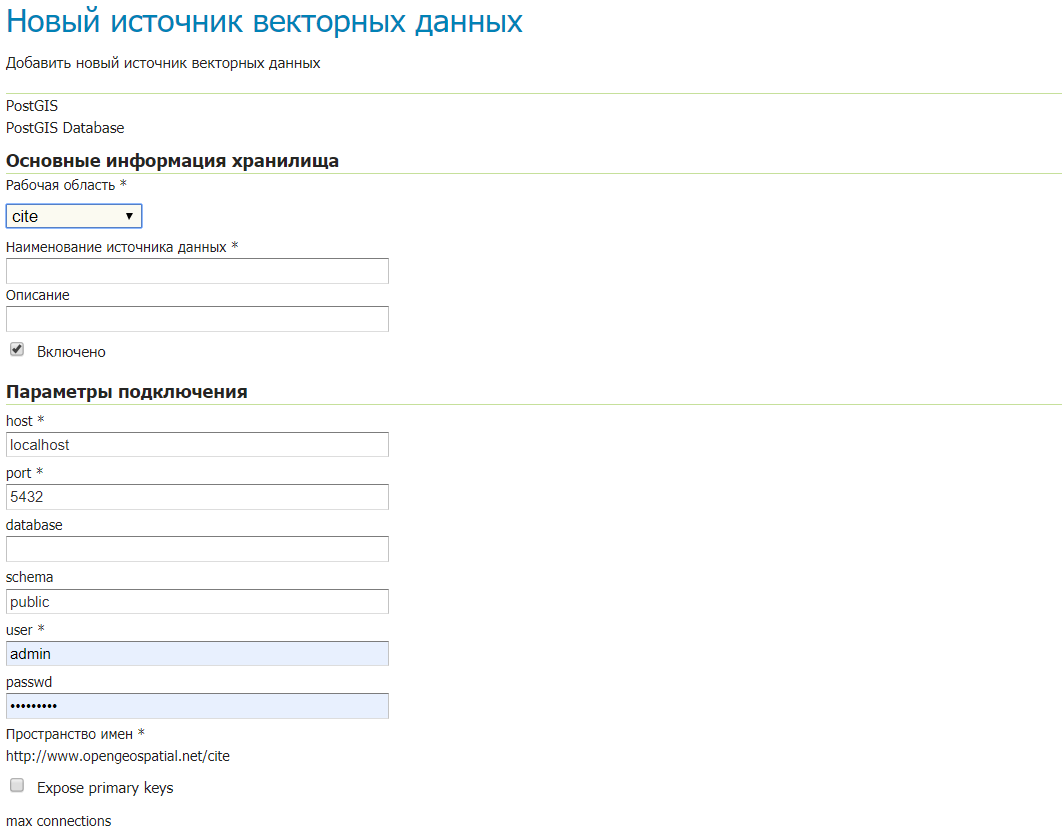


Рисунок 5.5 Выбор источника данных

Здесь необходимо выбрать рабочую область, к которой привязывается данный источник, а также указать в случае необходимости, другие параметры подключения, такие как логин, пароль, порт подключения и прочее.

5.2 Руководство пользователя веб-приложения

При открытии веб-приложения, пользователь попадает на главную страницу