2 Проектирование структуры приложения

2.1 Функциональное наполнение интернет-сервиса

Основные функции интернет-сервиса представлены на рисунке 2.1

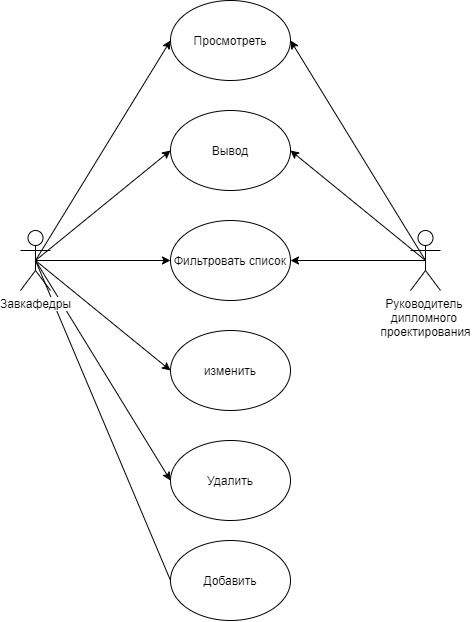


Рис.2.1 – Общая структура компонентов приложения

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 . 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).

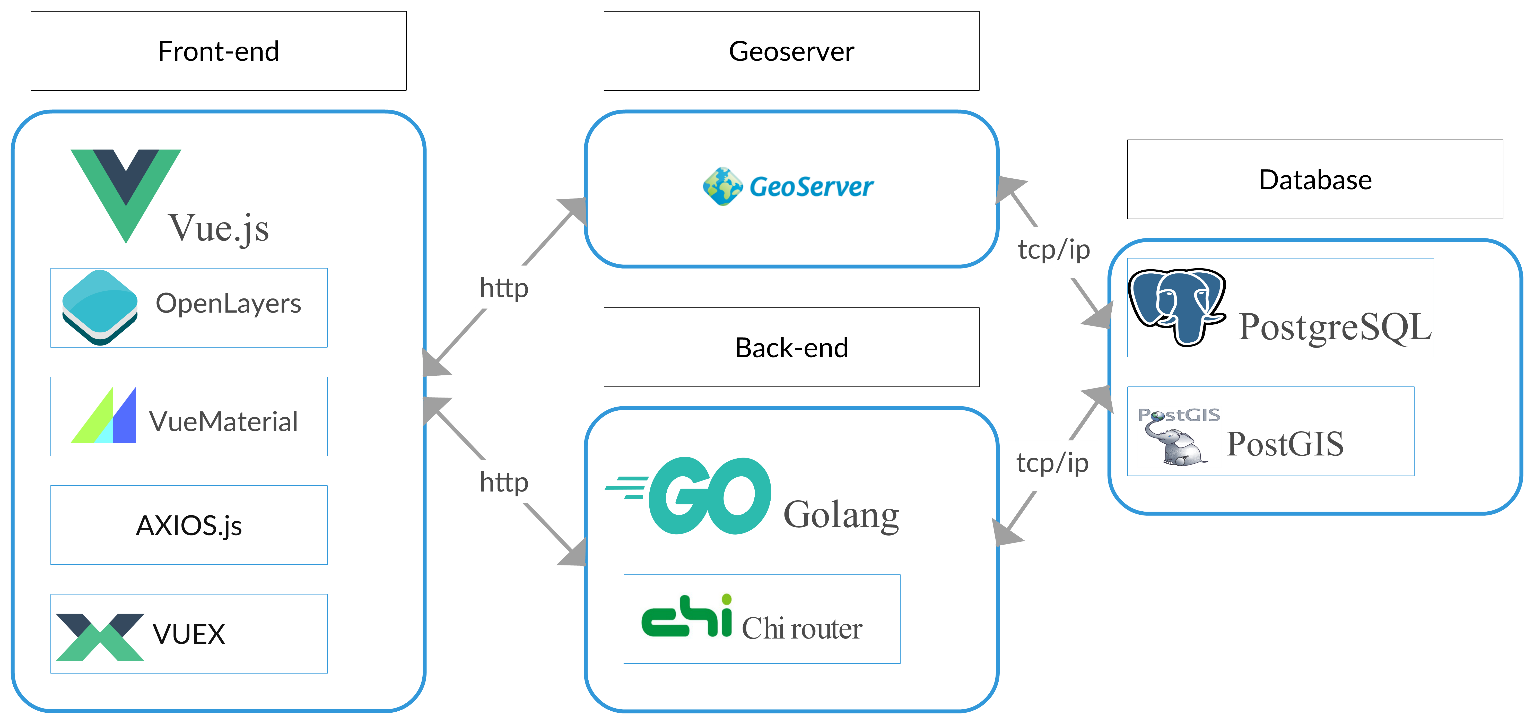


Рис.2.1 – Общая структура компонентов приложения

*Front-end* часть представлена веб-приложением, которое создано для кроссбраузерной работы по отображению 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.

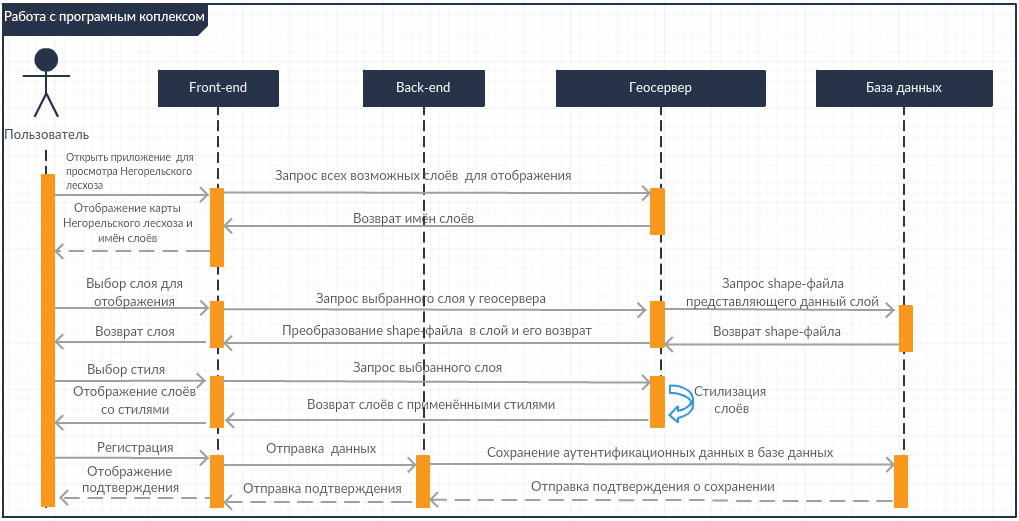


Рисунок 2.2 – Диаграмма последовательности

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 программного средства, которые являются так называемым «связующим звеном» между данными, хранящимися в базе данных и пользовательским веб-интерфейсом, который позволяет 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.

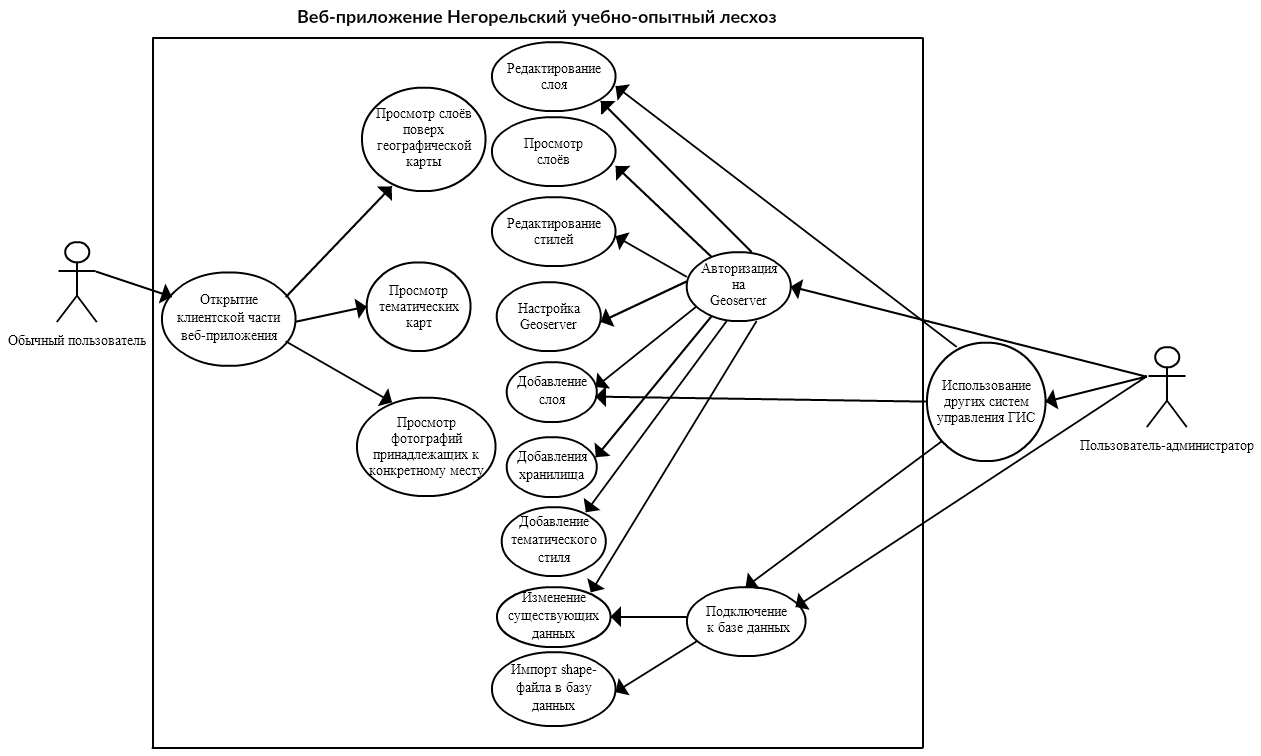


Рисунок 2.3 – Юз-кейс диаграмма приложения

В приложении существует два вида пользователей:

* Обычные пользователи
* Пользователи администраторы

Обычные пользователи имеют право только просматривать географическую и иную информацию, связанную с Негорельским учебно-опытным лесхозом. У администраторов же набор функций значительно больше. Они могут добавлять новую информацию в базу данных, выбирать для пользования уже существующие базы данных, редактировать уже имеющиеся данные или удалять их.

Основная информация об этих двух ролях представлена в таблице 2.1.

Таблица 2.1 — Виды ролей в приложении

|  |  |
| --- | --- |
| Роль | Назначение |
| Пользователь-администратор | Добавление, редактирование и удаление информации. Создание стилей для тематических карт. Управление базой данных и геосервером |
| Обычный пользователь | Пользование результатами взаимодействия базы данных, геосервера и бэк-энд части, представленными в веб-интерфейсе фронт-энд части. Просмотр карт, их тематическая раскраска, просмотр фотоданных, принадлежащих к какой-то точке |

2.2.1 Front-end

Front-end часть приложения спроектирована с применением VueJS, основные параметры приведены в таблице 2.2.

Таблица 2.2 — 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 |

2.2.2 Back-end

Back-end часть приложения спроектирована с применением Go, основные параметры приведены в таблице 2.3.

Таблица 2.3 — 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 |

2.2.3 База данных

В качестве базы данных используется PostgreSQL. Основные параметры приведены в таблице 2.4.

Таблица 2.4 — 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 2.2 представлена схема таблиц базы данных.

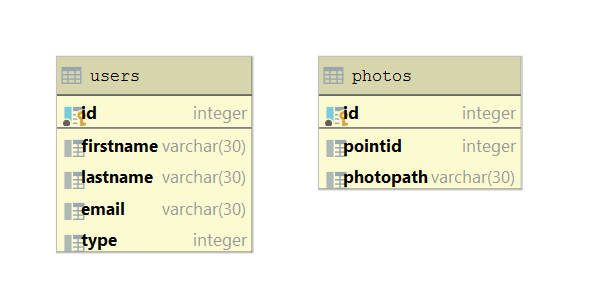


Рисунок 2.2 – Схема постоянных таблиц базы данных

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

Таблица 2.5 — Описание полей таблицы «users»

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

Продолжение таблицы 2.5

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

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

Таблица 2.6 — Описание полей таблицы «Photos»

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

2.2.4 Geoserver

В качестве центрального компонента приложения используется программное средство Geoserver. Основные параметры приведены в таблице 2.7.

Таблица 2.7 — 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, был сделан 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.

Для разработки серверной части приложения был выбран язык Go, так как он позволит значительно ускорить разработку, а также является надёжным средством для построения клиент-серверных приложений. Для разработки пользовательской части приложения был выбран VueJS фреймворк, так как он позволит разработать понятную и легко расширяемую структуру приложения в короткие сроки. В качестве «ядра» системы было выбрано программное средство Geoserver, которое идеально подходит для поставленных целей и задач.