Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

|  |  |  |
| --- | --- | --- |
| Согласовано |  | Утверждаю |
|  |  | Зав. кафедрой САПР и ПК |
|  |  |  |
|  |  | М.В. Щербаков |
| (подпись) (инициалы, фамилия) |  | (подпись) (инициалы, фамилия) |
| «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра

Разработка веб-сервиса для обработки данных мониторинга микроклимата и

(наименование работы)

экологии территории

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Автор |  |  | Островская Татьяна Сергеевна | |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) | |
| Обозначение | ВКР-40 461 806–10.27–23–24.81 | | |  |
|  | (обозначение документа) | | |  |
| Группа | ИВТ-463 |  | | |
|  | (шифр группы) |  | | |
| Направление | 09.03.01 «Информатика и вычислительная техника» | | | |
|  |  | | | |
|  | (код и наименование направления, наименование программы (профиля)) | | | |
| Руководитель работы |  |  | Парыгин Данила Сергеевич | |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) | |

Консультанты по разделам:

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |

Нормоконтролер Д.С. Парыгин

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2024 г.

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой САПР и ПК |
|  |  |
|  | М.В. Щербаков |
|  | (подпись) (инициалы, фамилия) |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Задание на | | | выпускную квалификационную работу бакалавра | | | | | | | |
|  | | | (наименование работы) | | | | | | | |
| Студент | | | Островская Татьяна Сергеевна | | | | | | | |
|  | | | (фамилия, имя, отчество) | | | | | | | |
| Код кафедры | | | 10.27 | | |  | | Группа | ИВТ-463 |  |
|  | | | | | |  | |  |  |  |
| Тема | | Разработка веб-сервиса для обработки данных мониторинга | | | | | | | | |
|  | | микроклимата и экологии территории | | | | | | | | |
| Утверждена приказом по ВолгГТУ | | | | | | | | | | |
| от | 01.11.2022 г. | | | № | 1522-ст | |  | | | |
|  | (дата подписания) | | |  | | |  | | | |
| Срок предъявления готовой работы | | | | | | | | 01 июня 2024 г. | |  |
|  | | | | | | | | (дата подписания) | |  |
| Исходные данные для выполнения работ | | | | | | | | | | |
| Материалы предшествующих исследований и перечень требований | | | | | | | | | | |
| по теме выпускной работы бакалавра, выданные научным руководителем | | | | | | | | | | |
| от кафедры САПРиПК | | | | | | | | | | |
|  | | | | | | | | | | |

Содержание основной части пояснительной записки

|  |  |
| --- | --- |
|  | Анализ предметной области и существующих решений |
|  | Проектирование программы |
|  | Реализация программы |
|  | Тестирование программы |
|  | Заключение |
|  | Список использованных источников |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Перечень графического материала

|  |  |
| --- | --- |
|  | Актуальность |
|  | Цели и задачи работы |
|  | Обзор существующих решений |
|  | Диаграмма вариантов использования |
|  | Архитектура решения |
|  | Описание алгоритмов и реализации |
|  | Средства разработки |
|  | Тестирование |
|  | Выводы |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель работы |  |  | Д.С. Парыгин |
|  | (подпись и дата подписания) |  | (инициалы и фамилия) |

Консультанты по разделам:

|  |  |  |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (краткое наименование раздела) | (подпись и дата подписания) | (фамилия, имя, отчество) |

Эта страница печатается на обороте предыдущей страницы!

Удалите эту надпись перед печатью документа!

Аннотация

Пояснительная записка к выпускной квалификационной работе Островской Т.С. по теме «Разработка веб-сервиса для обработки данных мониторинга микроклимата территории» описывает этапы разработки системы для обработки и визуализации данных мониторинга, полученных с аппаратного комплекса. В документе раскрываются подходы и технологии, примененные для решения задач подготовки данных для визуализации, реализации алгоритмов генерации тепловых карт и разработки программного продукта. Дипломная работа состоит из введения, четырех глав, заключения и списка литературы. Первая глава раскрывает основные понятия предметной области исследования и описывает аналогичные программные продукты. Во второй главе формулируется концепция разрабатываемой программы, приводится архитектура и прочие компоненты этапа проектирования. Третья глава описывает реализацию программы. Четвертая глава описывает тестирование разработанной программы.

Объем работы 57 листа, включающих 41 рисунок и 2 таблицы. При написании работы использовалось 24 источника. Ключевые слова: визуализация данных, мониторинг, микроклимат, веб-разработка.

Abstract

The explanatory note to the final qualifying work of T. S. Ostrovskaya on the topic “Development of web-service for processing of territory microclimate monitoring data” describes the stages of developing a system for processing and visualization of monitoring data obtained from the hardware complex. The document reveals the approaches and technologies applied to solving the problems of data preparation for visualization, implementation of heat map generation algorithms and software product development. Thesis consists of introduction, four chapters, conclusion and list of references. The first chapter reveals the basic concepts of the research subject area and describes similar software products. The second chapter formulates the concept of the application being developed, provides the architecture and other components of the design phase. The third chapter describes the implementation of the application. The fourth chapter describes the testing of the developed application.

The scope of work of 57 sheets, including 41 figures and 2 tables. 24 sources were used when writing the work. Keywords: data visualization, microclimate, monitoring, web development.

Содержание

[Список используемых сокращений 8](#_heading=h.gjdgxs)

[Введение 9](#_heading=h.30j0zll)

[1 Анализ предметной области и существующих решений 11](#_heading=h.1fob9te)

[1.1 Анализ предметной области](#_heading=h.1fob9te) мониторинга микроклимата территории [11](#_heading=h.1fob9te)

[1.2 Анализ существующих решений веб-сервисов](#_heading=h.3znysh7) для обработки данных мониторинга микроклимата [12](#_heading=h.3znysh7)

[1.2.1](#_heading=h.2et92p0) ArcGIS Online [12](#_heading=h.2et92p0)

[1.2.2 SmartCitizen 13](#_heading=h.tyjcwt)

[1.2.3](#_heading=h.3dy6vkm) EcoAtlac [14](#_heading=h.3dy6vkm)

[1.2.4 Результаты сравнения существующих решений 15](#_heading=h.1t3h5sf)

[2 Проектирование веб-сервиса 17](#_heading=h.4d34og8)

2.1 Информация для решения задачи [17](#_heading=h.2s8eyo1)

[2.2 Описание существующего процесса 17](#_heading=h.2s8eyo1)

[2.3](#_heading=h.17dp8vu) Описание автоматизированного процесса [18](#_heading=h.17dp8vu)

[2.4 Формулировка требований 19](#_heading=h.3rdcrjn)

[2.5 Анализ вариантов использования 20](#_heading=h.26in1rg)

[2.6 Архитектура веб-сервиса 22](#_heading=h.lnxbz9)

3 Реализация веб-сервиса для обработки и визуализации данных мониторинга микроклимата  [24](#_heading=h.35nkun2)

[3.1 Выбор средств разработки 24](#_heading=h.35nkun2)

[3.2](#_heading=h.1ksv4uv) Проектирование и реализация базы данных [28](#_heading=h.1ksv4uv)

[3.3](#_heading=h.44sinio) Разработка серверной части  [29](#_heading=h.44sinio)

[3.3.1](#_heading=h.44sinio) Алгоритм подготовки данных для визуализации  [29](#_heading=h.44sinio)

[3.3.2](#_heading=h.44sinio) Алгоритм проведения статистического анализа  [33](#_heading=h.44sinio)

[3.4](#_heading=h.2jxsxqh) Разработка клиентской части  [38](#_heading=h.2jxsxqh)

[3.4.1](#_heading=h.2jxsxqh) Разработка информационных карточек на главной странице  [38](#_heading=h.2jxsxqh)

[3.4.2](#_heading=h.2jxsxqh) Разработка формы загрузки файла  [39](#_heading=h.2jxsxqh)

[3.4.3](#_heading=h.2jxsxqh) Разработка секции для выбора параметров визуализации  [40](#_heading=h.2jxsxqh)

[3.4.4](#_heading=h.2jxsxqh) Разработка визуализации графика и таблицы с данными  [40](#_heading=h.2jxsxqh)

[3.4.5](#_heading=h.2jxsxqh) Разработка секции статистического анализа  [41](#_heading=h.2jxsxqh)

[3.4.6](#_heading=h.2jxsxqh) Разработка визуализации диаграммы и таблицы с данными  [41](#_heading=h.2jxsxqh)

[3.4.7](#_heading=h.2jxsxqh) Разработка визуализации тепловой карты  [42](#_heading=h.2jxsxqh)

4 Тестирование веб-сервиса  [43](#_heading=h.4i7ojhp)

4.1 Описание функционала веб-сервиса [43](#_heading=h.4i7ojhp)

[4.2 Входные и выходные данные 45](#_heading=h.2xcytpi)

[4.3 Функциональное тестирование 46](#_heading=h.1ci93xb)

[4.4 Тестирование отказоустойчивости 47](#_heading=h.3whwml4)

[4.5 Тестирование удобства интерфейса 48](#_heading=h.2bn6wsx)

[4.6](#_heading=h.2jxsxqh) Тестирование запросов на сервер  [49](#_heading=h.2jxsxqh)

[Заключение 53](#_heading=h.3as4poj)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 54](#_heading=h.1pxezwc)

[Приложение А Техническое задание 57](#_heading=h.49x2ik5)

Список используемых сокращений

БД – база данных.

API – Application Programming Interface (программный интерфейс приложения).

REST – Representational State Transfer (передача состояния представления).

SQL – Structured Query Language (язык структурированных запросов).

UI – User Interface (пользовательский интерфейс).

DB – Data Base (база данных).

HTTP-запросы – HyperText Transfer Protocol (протокол передачи гипертекста).

SVG – Scalable Vector Graphics (масштабируемая векторная графика).

HTML – HyperText Markup Language (язык гипертекстовой разметки).

CSS – Cascading Style Sheets (каскадные таблицы стилей).

CSV – Comma-Separated Values (значения, разделённые запятыми).

JSON – JavaScript Object Notation (текстовый формат обмена данными, основанный на JavaScript).

ERD – Entity-Relationship diagram (модель «сущность — связь»).

URL – Uniform Resource Locator (единообразный указатель местонахождения ресурса).

Введение

В современном мире вопросы жилищного строительства и благоустройства территорий становятся все более актуальными и важными для обеспечения комфортного проживания граждан. Согласно статистике Росстата, за 2023 год в России было введено жилья общей площадью 110,44 миллиона квадратных метров, что является рекордным показателем и превышает результаты предыдущего года на 7,4% [1]. Этот значительный рост отражает стремление к улучшению жилищных условий и расширению доступного жилья для населения. Одним из ключевых факторов, влияющих на качество застройки, является оценка микроклимата территории. Эффективный мониторинг позволяет контролировать и анализировать параметры окружающей среды, такие как температура, влажность, скорость ветра и другие, что важно для создания комфортной и безопасной городской среды. В данном контексте разработанный программный продукт играет ключевую роль, предоставляя пользователям возможность просматривать информацию о микроклимате в удобном формате. Благодаря различным графикам и тепловым картам пользователи могут получить наглядное представление о состоянии микроклимата на территории застройки, что позволяет принимать обоснованные решения по улучшению окружающей среды и повышению качества жизни граждан.

В связи с этим, целью работы являлась разработка программного продукта, который позволит осуществлять анализ и визуализацию данных пространственно-распределённого мониторинга микроклимата, обеспечивая принятие обоснованных решений при планировании и благоустройстве застроенных территорий.

Для достижения поставленной цели необходимо было решить следующие задачи:

1) провести анализ методов визуализации и обработки данных о микроклимате с использованием географической привязки;

2) провести анализ существующих решений в области мониторинга и визуализации данных о микроклимате;

3) спроектировать веб-модуль для визуализаций данных о микроклимате и представления статистики на выбранных территориях;

4) разработать и протестировать веб-модуль для визуализации данных о микроклимате и представления статистики на выбранных территориях.

1 Анализ предметной области и существующих решений

1.1 Анализ предметной области мониторинга микроклимата территории

Мониторинг микроклимата играет важную роль в различных областях, таких как строительство, урбанистика, сельское хозяйство и экология. В условиях современных экологических вызовов и активного градостроительства, информация о микроклимате необходима для эффективного планирования и управления территориями. На протяжении многих лет мониторинг микроклимата эволюционировал от использования простых аналоговых приборов до применения сложных систем автоматизированного сбора данных. Ранее специалисты пользовались термометрами, гигрометрами и другими ручными приборами для сбора микроклиматических данных, что требовало значительных временных и трудовых затрат и не позволяло получать информацию в реальном времени.

С развитием технологий и появлением цифровых датчиков началась автоматизация процессов сбора данных. Электронные устройства значительно упростили процесс измерений, повысили точность и позволили интегрировать данные с компьютерными системами для дальнейшего анализа. Появление Интернета вещей (IoT) стало следующим шагом в эволюции мониторинга микроклимата, позволив подключать датчики к сети и собирать данные в режиме реального времени [2]. Это открывает новые возможности для анализа и управления микроклиматическими условиями, а также обеспечивает более оперативное реагирование на изменения.

Современные информационные технологии предлагают мощные инструменты для работы с большими объемами данных, что особенно актуально для мониторинга микроклимата. Географические информационные системы (ГИС) позволяют не только собирать и хранить данные, но и визуализировать их на интерактивных картах, что облегчает понимание и анализ информации [3]. Существующие решения в области мониторинга микроклимата разнообразны и включают как простые системы для домашнего использования, так и комплексные решения для профессионального применения. Однако многие из них имеют ограничения, связанные с узкой специализацией или недостаточной гибкостью в настройке и интеграции данных.

Необходимость создания универсального веб-сервиса для мониторинга микроклимата обусловлена потребностью в систематическом и доступном инструменте, который бы позволял собирать, анализировать и визуализировать данные из различных источников. Такой сервис должен быть адаптирован для использования различными специалистами, включая застройщиков, экологов, агрономов и управляющих парками. Важным аспектом является обеспечение наглядности представления данных, что облегчает их интерпретацию и принятие управленческих решений. Таким образом, создание веб-сервиса для обработки данных микроклимата является актуальной задачей, требующей интеграции современных технологий и учета потребностей пользователей. В следующем разделе будет проведен анализ существующих решений, что позволит выделить ключевые требования и определить направление дальнейшей разработки.

1.2 Анализ существующих решений веб-сервисов для обработки данных мониторинга микроклимата

1.2.1 ArcGIS Online

ArcGIS Online представляет собой платформу для создания, анализа и публикации карт и пространственных данных в интернете [4]. Он предоставляет широкий спектр функциональных возможностей, включая создание интерактивных карт, проведение анализа пространственных данных, интеграцию с различными источниками данных и API, доступ к базам данных гео-пространственной информации и другие возможности. Пользователи могут создавать собственные карты, добавлять на них слои данных, проводить анализ пространственной информации, а также публиковать их в интернете для общего доступа или ограниченного круга лиц. На рисунке 1 представлен экран сервиса ArcGIS.

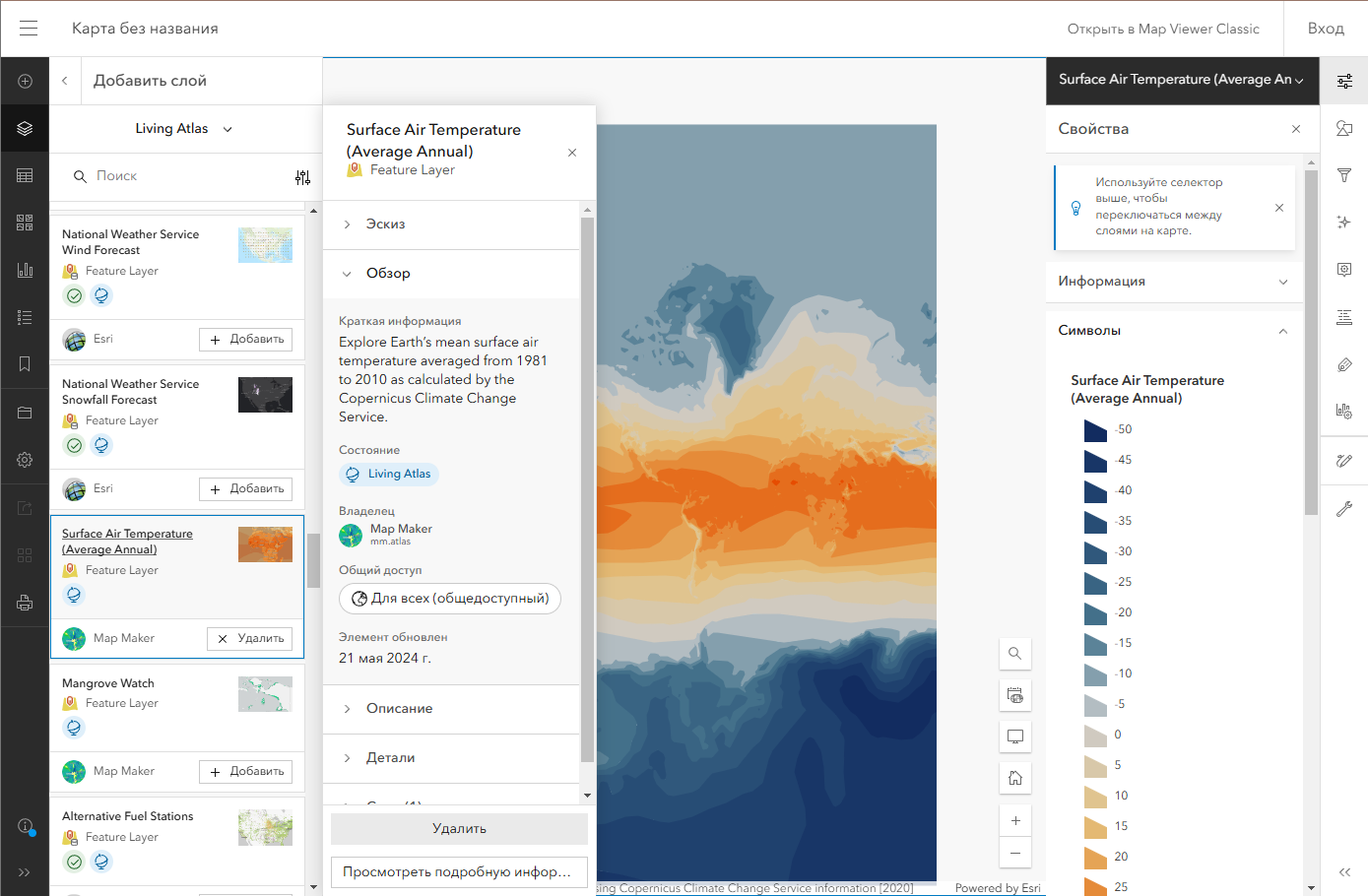


Рисунок 1 – Пример работы веб-сервиса ArcGIS Online

1.2.2 Smart Citizen

Smart Citizen — это городская платформа для мониторинга окружающей среды и улучшения качества жизни в городах [5]. Она предоставляет пользователям возможность устанавливать датчики на своих домах или в общественных местах для сбора данных о качестве воздуха, уровне загрязнения, уровне шума и других параметрах окружающей среды. Собранные данные передаются в облачное хранилище, где они анализируются и визуализируются на картах для общего доступа. Пользователи могут просматривать данные, а также принимать меры по улучшению экологической ситуации в своем городе. На рисунке 2 представлен экран сервиса Smart Citizen.

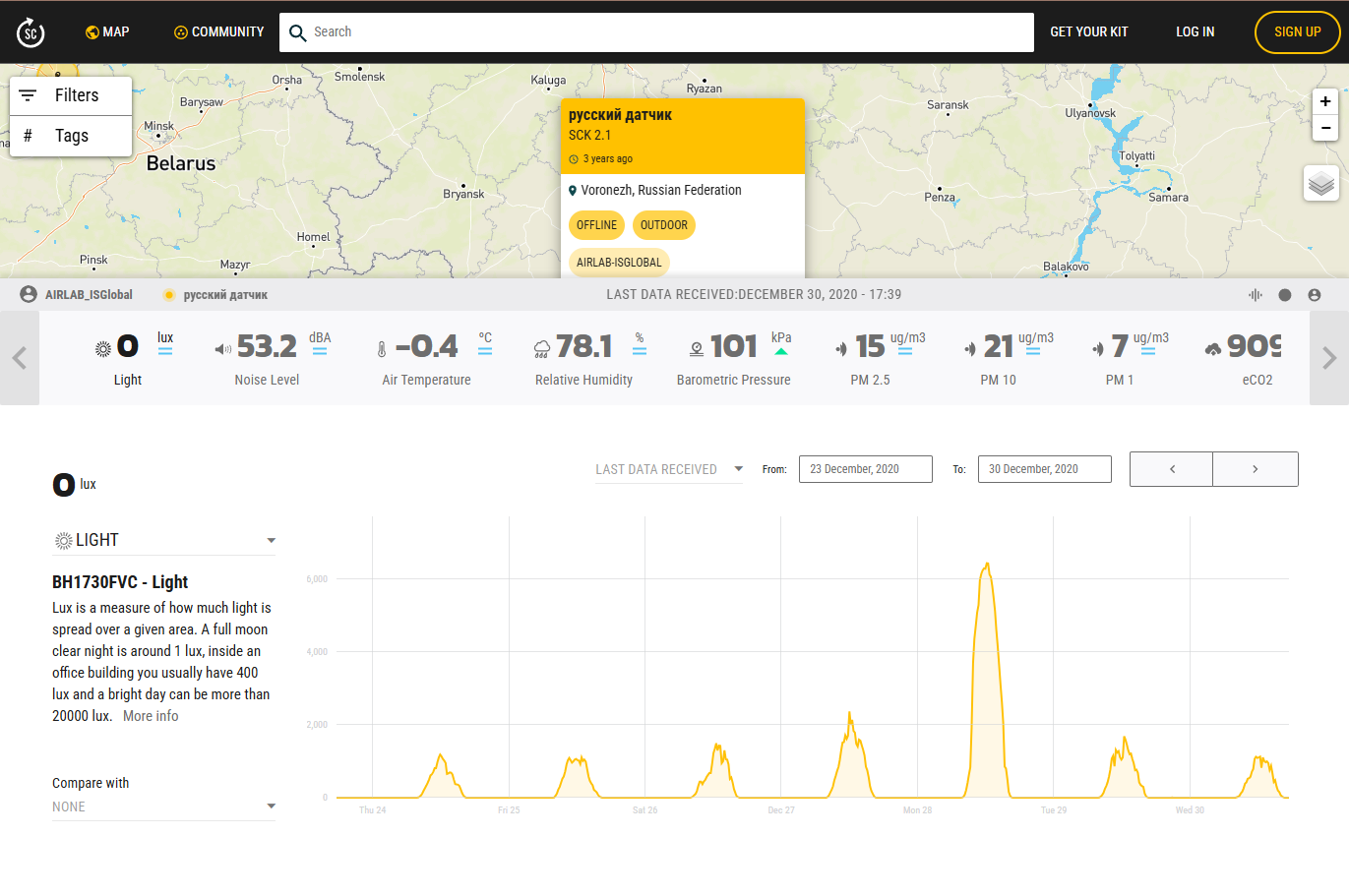


Рисунок 2 – Пример работы веб-сервиса Smart Citizen

1.2.3 EcoAtlas

EcoAtlas — это онлайн-сервис, предоставляющий информацию о состоянии окружающей среды и ее воздействии на здоровье человека [6]. Он содержит данные о качестве воздуха, воды, почвы, уровне шума, выбросах вредных веществ и других параметрах окружающей среды в различных регионах. Пользователи могут просматривать интерактивные карты с данными, анализировать информацию, получать уведомления о превышении нормативов и участвовать в мониторинге экологической ситуации в своем регионе. На рисунке 3 представлен экран сервиса EcoAtlas.

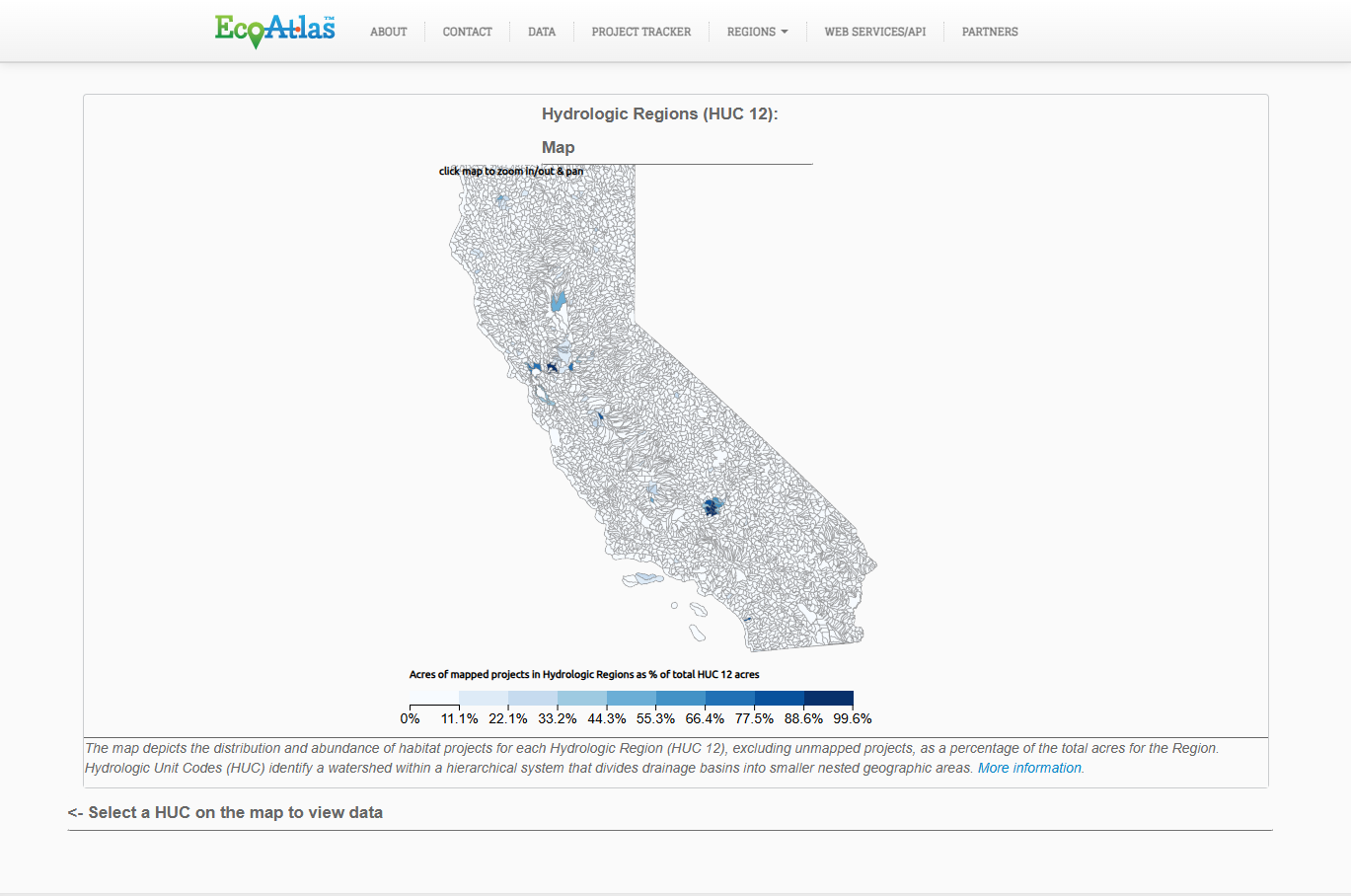


Рисунок 3 – Пример работы веб-сервиса EcoAtlas

1.2.4 Результаты сравнения существующих решений

Проведем итоговое сравнение существующих решений задачи построения рейтинга недвижимости по следующим критериям:

* интерфейс пользователя;
* аналитические возможности;
* возможности интеграции;
* количественные показатели данных;
* качество визуализации данных;
* географическое покрытие;
* доступность API.

Результаты проведенного сравнительного анализа вышеописанных решений представлены в таблице 1.

Таблица 1 – Результаты сравнения существующих решений

| Критерий\решение | ArcGIS Online | SmartCitizen | ЭкоАтлас |
| --- | --- | --- | --- |
| Возможность загрузки своих данных | + | – | + |
| Возможность генерации тепловой карты | + | + | + |
| Выявление аномалий | + | – | + |
| Возможность интеграции с другими сервисами | + | + | + |
| Поддержка русского языка | – | + | + |
| Возможность построения прогнозов | + | – | – |
| Просмотр статистики и аналитики | – | + | + |

2 Проектирование веб-сервиса

2.1 Информация для решения задачи

Разрабатываемый проект состоит из двух основных частей: аппаратного комплекса для сбора данных мониторинга и веб-сервиса для визуализации

собранных данных. Аппаратный комплекс разработан другим студентом в рамках выпускной работы. На стороне аппаратной части, комплекс включает в себя датчики для измерения параметров микроклимата, такие как температура воздуха, его влажность, уровень загрязнения и другие. Для обработки и передачи данных на серверную часть пользователь загружает данные на носитель информации и в последствии загружает их в БД веб-сервиса.

Клиентская часть приложения включает в себя графический интерфейс пользователя для визуализации и анализа данных. Пользователь может видеть графики, диаграммы и таблицы, отображающие статистические данные за определенный период времени.

Таким образом, программно-аппаратный комплекс с приложением для обработки и анализа данных о микроклиматических условиях будет обеспечивать надежное измерение параметров микроклимата и предоставляет пользователю удобный способ визуализации и анализа данных. Общая архитектура проекта представлена на рисунке 4.

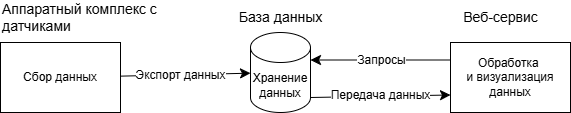


Рисунок 4 – Общая архитектура проекта

2.2 Описание существующего процесса

Сейчас для мониторинга микроклимата территории застройки требуется собрать данные и вручную обрабатывать: очищать, структурировать и агрегировать. Однако, помимо обработки данных, необходимо отдельно визуализировать информацию для анализа и принятия решений. Процесс включает в себя использование различных инструментов для анализа и создания визуализаций, что требует значительных временных и трудовых затрат. Диаграмма существующего процесса [7] приведена на рисунке 5.

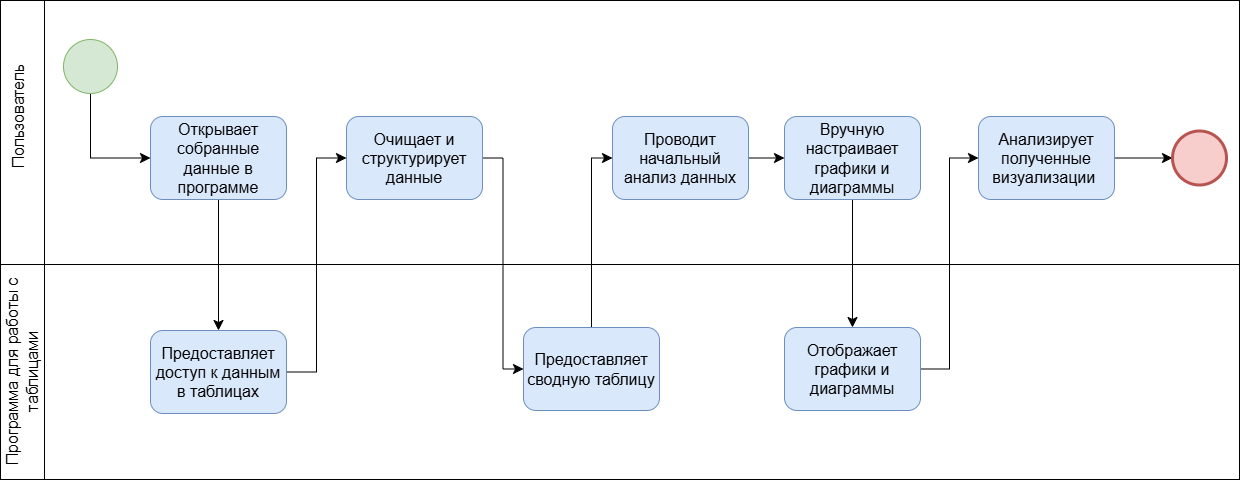


Рисунок 5 – Диаграмма as-is

Проанализировав диаграмму, можно сделать вывод, что весь этот процесс требует значительных временных и трудовых затрат, а также навыков работы с различными программами для обработки и визуализации данных.

2.3 Описание автоматизированного процесса

Проектируемый процесс призван упростить и ускорить мониторинг микроклимата территории, а также сэкономить ресурсы заинтересованных лиц. Диаграмма автоматизированного процесса [7] представлена на рисунке 6. Сравнив диаграммы на рисунке 5 и рисунке 6, можно заметить, что проектируемый веб-сервис позволит компаниям и организациям избежать ручной обработки и визуализации данных. Веб-сервис автоматизирует весь процесс: данные после поступления в систему автоматически обрабатываются и визуализируются в виде удобных графиков и тепловых карт. Это значительно упрощает анализ и принятие решений.

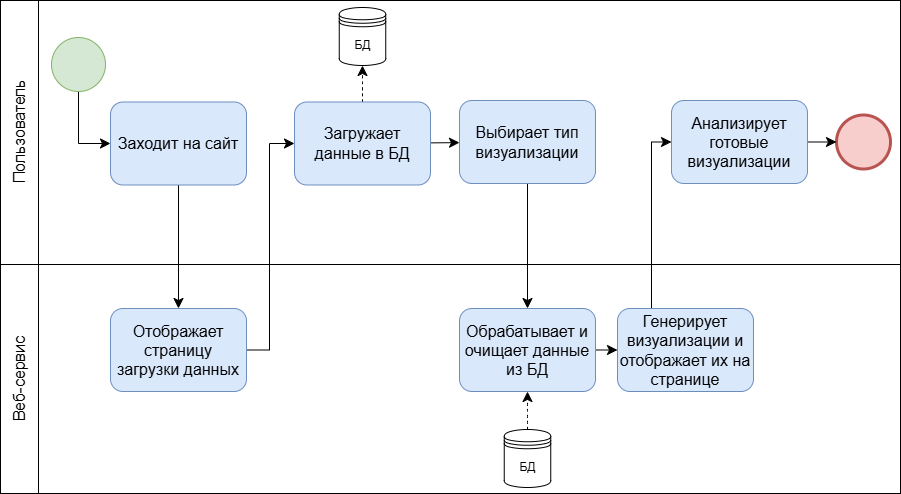


Рисунок 6 – Диаграмма to-be

2.4 Формулировка требований

В результате анализа предметной области, а также в соответствии с поставленными требованиями был сформирован перечень требований к разрабатываемому мобильному приложению:

* веб-сервис должен позволять осуществлять загрузку данных из аппаратного комплекса и их визуализацию на сетевом ресурсе;
* должна быть возможность выполнять загрузку данных с флэш-накопителя, агрегирующего данные датчиков мониторинга микроклимата и экологии воздуха;
* должна быть реализована база для хранения всех загружаемых данных;
* загруженные в сервис данные должны подготавливаться для визуализации;
* визуализация данных выполняется в виде графиков, позволяющих просмотр собираемых показателей микроклимата и экологии за настраиваемые промежутки времени;
* при использовании параметров геопозиционирования собираемых показателей должна быть реализована возможность их отображения в виде тепловой карты, цвето-габаритные характеристики которой отражают свойства визуализируемых данных (величину показателя в точке).

2.5 Анализ вариантов использования

Диаграмма вариантов использования [8] представлена на рисунке 7. Рассмотрим возможных акторов проектируемого приложения. На приведенной ниже use-case диаграмме перечислены следующие функции:

* изучение информации о микроклимате;
* загрузка данных для мониторинга;
* генерация графика;
* генерация диаграммы;
* генерация тепловых карт;
* выбор параметров визуализации;
* проведение статистического анализа;
* выявление аномалий.

Изучение информации о микроклимате — процесс визуального ознакомления пользователя с данными о микроклимате на территории. Доступно для всех пользователей, которые могут просматривать и анализировать представленные данные для получения необходимой информации.

Загрузка данных для мониторинга — процесс, в котором пользователь загружает новые данные о микроклимате в систему.

Генерация графика — процесс создания графиков для визуализации данных о микроклимате. Пользователь выбирает необходимые параметры, и система автоматически генерирует соответствующий график, который позволяет лучше понять тренды и изменения в микроклимате.

Генерация диаграммы — процесс создания диаграмм для визуализации данных о микроклимате. Пользователь выбирает параметры для диаграммы, и система генерирует визуальное представление данных, что помогает в анализе и принятии решений.

Генерация тепловых карт — процесс создания тепловых карт, которые позволяют визуализировать распределение температурных и других микроклиматических параметров на территории. Пользователь выбирает параметры, и система создает тепловую карту, что позволяет легко определить аномальные зоны и особенности микроклимата.

Выбор параметров визуализации — процесс настройки параметров визуализации для графиков, диаграмм и тепловых карт. Пользователь выбирает необходимые параметры, такие как временные рамки, тип параметра и другие настройки, чтобы настроить визуализацию под свои нужды.

Проведение статистического анализа данных о микроклимате — процесс, в котором пользователь выполняет статистический анализ собранных данных о микроклимате. Это позволяет выявить тренды, корреляции и другие статистически значимые зависимости в данных.

Выявление аномалий — процесс анализа данных для обнаружения аномальных значений и событий в микроклимате. Пользователь использует систему для автоматического выявления отклонений от нормы, что помогает в раннем обнаружении потенциальных проблем и принятии необходимых мер.

Каждый из перечисленных вариантов использования играет важную роль в обеспечении эффективного мониторинга и анализа микроклимата территории, что позволяет пользователям получать актуальную информацию и принимать обоснованные решения на основе полученных данных.

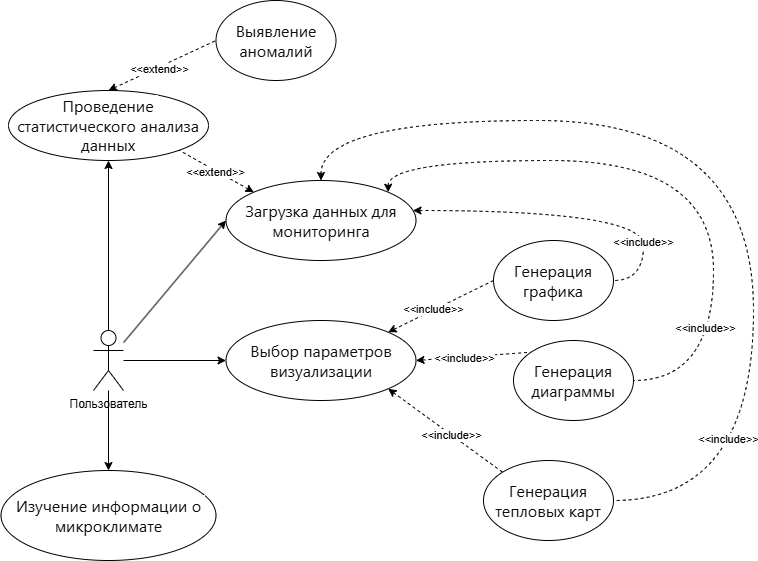


Рисунок 7 – Use-case диаграмма вариантов использования проектируемого программного решения

2.6 Архитектура веб-сервиса

Для реализации веб-сервиса была выбрана клиент-серверная архитектура [9]. Данный подход удобен тем, что позволяет разделить логику приложения на две части: клиентскую и серверную, что обеспечивает более гибкое управление и масштабирование. Клиентская часть реализована с использованием фреймворков Vue 3 [10] и Quasar [11], что позволяет создавать интерактивный и удобный пользовательский интерфейс. Серверная часть построена на базе FastAPI [12], что обеспечивает высокую производительность и простоту разработки API.

Общим архитектурным подходом для реализации веб-сервиса был выбран принцип разделения на клиентскую и серверную части. Схема этой архитектуры представлена на рисунке 8. Как видно из схемы, архитектура делится на несколько слоев, где каждый слой отвечает за определенные функции системы.

Клиентская часть - это слой, отвечающий за взаимодействие пользователя с системой. Компонент UI отвечает за отрисовку данных и обработку действий пользователя. Использование Vue 3 [10] и Quasar [11] обеспечивает высокую интерактивность и гибкость интерфейса, позволяя пользователям легко просматривать и настраивать визуализацию данных.

Серверная часть - это слой, отвечающий за обработку запросов от клиента и взаимодействие с базой данных. Компонент API, реализованный на FastAPI [12], обрабатывает HTTP-запросы, выполняет необходимую бизнес-логику и взаимодействует с базой данных PostgreSQL [13]. FastAPI обеспечивает высокую производительность и поддержку асинхронных операций, что особенно важно для работы с большими объемами данных.

База данных - это слой, отвечающий за хранение и управление данными о микроклимате. PostgreSQL используется в качестве основного хранилища данных, что обеспечивает надежное и масштабируемое управление данными. Компонент DB обеспечивает доступ к данным, их обновление и выполнение сложных запросов для анализа.

Проектируемый процесс призван ускорить обработку и визуализацию данных микроклимата, а также сэкономить ресурсы пользователей. Использование клиент-серверной архитектуры позволяет эффективно разделить задачи между клиентом и сервером, обеспечивая высокую производительность и удобство использования.

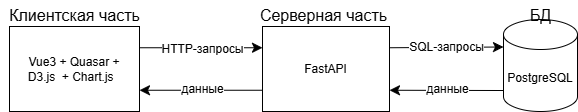


Рисунок 8 – Схема общей архитектуры веб-сервиса

3 Реализация веб-сервиса для обработки и визуализации данных мониторинга микроклимата

3.1 Выбор средств разработки

В качестве языка программирования для разработки серверной части был выбран язык Python [14]. Python — это высокоуровневый язык программирования, разработанный Гвидо ван Россумом и выпущенный впервые в 1991 году. Благодаря своей простоте и читабельности, Python является одним из самых популярных языков программирования, что обеспечивает большое количество ресурсов, документации и активное сообщество разработчиков. По сравнению с альтернативами, такими как Java или C++, Python обеспечивает более быстрое прототипирование и разработку, поэтому является оптимальным выбором для создания веб-сервиса визуализации данных.

Для реализации интерфейса приложения был выбран JavaScript-фреймворк Vue 3 [10]. Впервые представленный в сентябре 2020 года, Vue 3 предлагает значительные улучшения по сравнению с предыдущей версией, включая лучшую производительность, улучшенную поддержку TypeScript и новую Composition API для более гибкой и масштабируемой разработки компонентов. Vue 3 призван решить проблемы, связанные с масштабируемостью и сложностью кода, предлагая более интуитивный и удобный подход к созданию пользовательских интерфейсов.

Для более эффективного и быстрого создания пользовательских интерфейсов был также выбран Quasar Framework [11], который построен на основе Vue.js и предлагает обширный набор компонентов и инструментов. Quasar предоставляет мощные инструменты для разработки и сборки приложений, минимизируя время разработки и обеспечивая высокое качество пользовательских интерфейсов. Этот фреймворк упрощает процесс разработки, устраняя необходимость в многочисленных сторонних библиотеках, и позволяет создавать приложения с единой кодовой базой для различных платформ.

Для взаимодействия с REST API была выбрана библиотека FastAPI [12]. FastAPI — это современный, высокопроизводительный веб-фреймворк для Python [14], который позволяет создавать API с высокой скоростью и эффективностью. FastAPI использует асинхронные вызовы для обработки запросов, что значительно увеличивает производительность и снижает задержки. Он поддерживает автоматическую генерацию документации API, что упрощает процесс разработки и тестирования. Кроме того, FastAPI обеспечивает строгую типизацию данных, что повышает надежность и безопасность кода, позволяя избежать многих ошибок на этапе разработки. Фреймворк также легко интегрируется с различными базами данных и внешними сервисами, что делает его идеальным выбором для создания высоконагруженных и масштабируемых веб-приложений.

Для хранения данных в проекте выбрана PostgreSQL [13]. Это мощная и надежная реляционная база данных с открытым исходным кодом, обладающая высокой производительностью и поддержкой стандартов SQL. PostgreSQL обеспечивает безопасность данных и гибкость в настройке, что делает ее идеальным выбором для разработки приложений любого уровня сложности.

Для визуализации данных была выбрана библиотека Chart.js [15]. Chart.js используется для создания простых и анимированных графиков. Она поддерживает различные типы графиков, такие как линейные, столбчатые, круговые, радиальные и другие. Библиотека проста в использовании и предоставляет возможность настраивать внешний вид графиков, добавлять анимации, легенды и всплывающие подсказки. Chart.js также поддерживает возможность работы с большими объемами данных и обеспечивает высокую производительность благодаря использованию HTML5 Canvas.

Для создания более сложных и интерактивных визуализаций была выбрана библиотека D3.js [16]. D3.js используется для манипуляции данными и создания динамических, интерактивных визуализаций с использованием стандартов веб-технологий, таких как SVG, HTML и CSS. Она предоставляет мощные средства для работы с данными, включая функции для масштабирования, осей, анимаций и обработки событий. D3.js позволяет создавать сложные графики и визуализации, которые могут обновляться в реальном времени и реагировать на действия пользователя, что делает её незаменимым инструментом для визуализации данных о микроклимате и экологии.

Для обработки и анализа данных была выбрана библиотека pandas [17]. pandas предоставляет высокоуровневые структуры данных, такие как DataFrame, которые позволяют эффективно манипулировать данными в табличной форме. С её помощью можно выполнять операции по фильтрации, агрегированию, объединению и трансформации данных. pandas поддерживает работу с данными из различных источников, включая CSV, Excel, SQL и JSON, что делает её полезным инструментом для анализа и предобработки данных.

Для численных вычислений и работы с массивами данных используется библиотека numpy [18]. numpy предоставляет мощные возможности для работы с многомерными массивами и матрицами, а также широкий набор математических функций для выполнения операций над ними. Эта библиотека оптимизирована для высокопроизводительных вычислений, что делает её хорошим выбором для обработки больших объемов данных и выполнения сложных математических вычислений, таких как статистический анализ и обработка сигналов.

Библиотека csv [19] используется для чтения и записи данных в формате CSV. Она является встроенной в Python и предоставляет простой способ обработки текстовых данных, организованных в таблицы. С помощью этой библиотеки можно легко парсить CSV-файлы, извлекать из них данные и сохранять их в другие форматы. csv удобна для работы с большими объемами данных, которые нужно быстро и эффективно обработать.

Для взаимодействия с базой данных в проекте была использована библиотека SQLAlchemy [20]. SQLAlchemy является мощным инструментом для работы с базами данных, который поддерживает объектно-реляционное отображение (ORM). Это позволяет разработчикам работать с базой данных, используя объектно-ориентированный подход, что упрощает написание и чтение кода.

Для контроля версий был выбран Git [21]. Git используется для управления версиями исходного кода и совместной работы над проектами. Он предоставляет инструменты для отслеживания изменений, создания веток, слияния и восстановления версий кода. Git поддерживает распределённую архитектуру, что позволяет каждому разработчику работать с локальной копией репозитория и синхронизировать изменения с центральным репозиторием по мере необходимости. Git обеспечивает надёжное хранение истории изменений, удобные механизмы разрешения конфликтов и интеграцию с различными инструментами для повышения эффективности разработки.

В качестве среды разработки были выбраны Visual Studio Code [22] и PyCharm [23]. Visual Studio Code — это бесплатный редактор кода с открытым исходным кодом от Microsoft. Он предоставляет множество расширений для различных языков программирования, поддержку отладки, интеграцию с системами контроля версий и возможности для работы с удалёнными серверами. Visual Studio Code отличается лёгкостью, высокой производительностью и возможностью настройки под индивидуальные потребности разработчика.

PyCharm [23] — это интегрированная среда разработки (IDE) от JetBrains, специально созданная для разработки на Python [14]. Она поддерживает продвинутую автодополняемость кода, удобные инструменты для отладки, тестирования и работы с базами данных. PyCharm также включает в себя инструменты для рефакторинга кода, проверки соответствия стандартам кодирования и интеграцию с системами контроля версий. Эта среда разработки обеспечивает высокую продуктивность и удобство работы благодаря своим мощным инструментам и удобному интерфейсу.

3.2 Проектирование и реализация базы данных

Таблица «climate\_date» предназначена для хранения основных данных о мониторинге микроклимата. Она содержит информацию о времени измерения, показателях температуры, влажности и уровне загрязнения, а также координатах местоположения, где были проведены измерения. Каждая запись в таблице представляет отдельное измерение микроклимата в определенный момент времени и месте. ERD-диаграмма представлена на рисунке 9.

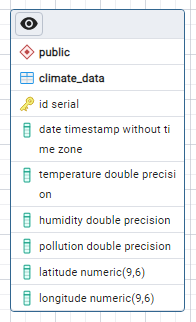


Рисунок 9 – ERD-диаграмма

Таблица имеет следующие столбцы:

* id: уникальный идентификатор записи. Каждая запись в таблице имеет уникальный числовой идентификатор, который автоматически увеличивается при добавлении новых записей (тип данных serial);
* date: дата и время снятия показаний микроклимата. Для каждой записи указывается дата, когда были получены данные (тип данных timestamp);
* temperature: температура воздуха в градусах Цельсия. Значение представлено в виде числа с плавающей точкой (тип данных decimal);
* humidity: влажность воздуха в процентах. Числовое значение, представленное в виде десятичной дроби (тип данных decimal);
* pollution: концентрация углекислого газа в воздухе в частях на миллион (ppm). Значение также представлено в виде числа с плавающей точкой (тип данных decimal);
* latitude: географическая широта места измерения микроклимата. Это числовое значение, представленное в виде десятичной дроби (тип данных decimal);
* longitude: географическая долгота места измерения микроклимата. Также числовое значение, представленное в виде десятичной дроби (тип данных decimal).

3.3 Разработка серверной части

3.3.1 Алгоритм подготовки данных для визуализации

Для различных типов визуализаций применяются разные методы подготовки данных. В нашем случае используются три вида визуализаций: график, диаграмма и тепловая карта. Рассмотрим процесс подготовки данных для каждого вида.

Для визуализации графиков данные проходят несколько этапов обработки. На первом этапе с клиентской части системы передается запрос. Первоначально данные фильтруются по заданным параметрам, таким как тип параметра (температура, влажность, загрязнение), диапазон дат и период агрегации (дни, месяцы, годы). Фильтрация осуществляется с помощью SQL-запроса с параметрами визуализации. После фильтрации данные могут быть агрегированы по дням, месяцам или годам. Это упрощает анализ временных рядов и позволяет пользователю видеть обобщенную информацию за указанные периоды. Если выбранный период агрегации — дни, то группировка не требуется, и данные остаются в исходном виде. Для группировки по месяцам и годам используется метод resample библиотеки pandas [17], который позволяет агрегировать данные по указанным временным интервалам. Полный цикл подготовки данных для графика представлен на рисунке 10.

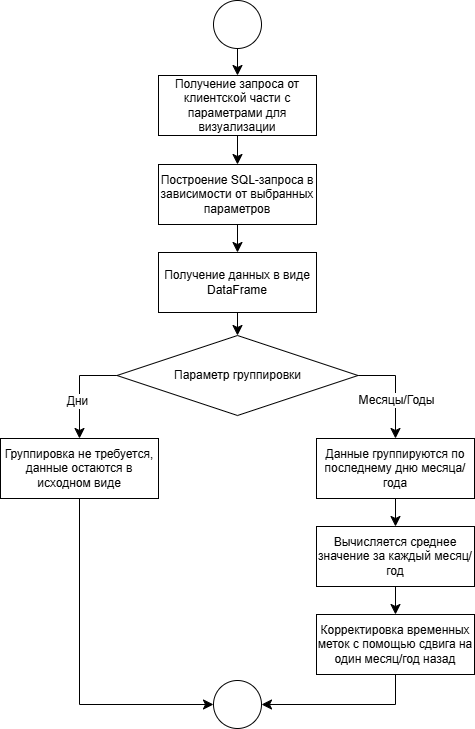


Рисунок 10 – Алгоритм подготовки данных для визуализации типа «график»

Фильтрация и группировка данных перед визуализацией позволяют обрабатывать и анализировать большие объемы данных, получая обобщенные и удобные для восприятия результаты.

Подготовка данных для визуализации диаграмм распределения значений включает несколько шагов: округление значений, группировка данных и подсчет частоты. Эти шаги помогают преобразовать сырые данные в формат, удобный для создания диаграмм и анализа распределений. Сначала значения округляются до ближайшего 0.5 единицы, что помогает упрощать данные и уменьшить количество уникальных значений, что делает диаграмму более понятной. После округления данных необходимо сгруппировать их по значению. Это делается с использованием метода groupby библиотеки pandas [17]. Результатом этого шага является DataFrame, где каждая строка представляет уникальное значение value и его частоту frequency. Для подготовки данных к отправке для визуализации необходимо преобразовать DataFrame в подходящий формат. В данном случае создается список точек, каждая из которых представлена словарем с ключами x (значение параметра) и y (частота появления этого значения в данных). Подробный процесс представлен на рисунке 11.

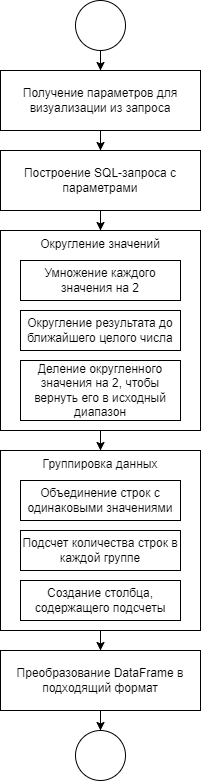


Рисунок 11 – Алгоритм подготовки данных для визуализации типа «диаграмма»

Процесс подготовки данных для тепловых карт включает несколько этапов, которые обеспечивают корректное отображение пространственных данных с учетом заданных параметров. Данные извлекаются из базы данных на основе переданных параметров. Используется SQL-запрос, который фильтрует данные по географическим координатам и заданному параметру. После получения данных из базы, они обрабатываются для последующего использования в визуализации. Обработка включает расчет минимальные и максимальные значения для широты и долготы, чтобы установить границы области отображения тепловой карты. Эти значения используются для масштабирования тепловой карты и обеспечения правильного отображения всех точек данных. Для формирования ответа данные преобразуются в формат, подходящий для визуализации тепловой карты. Каждая точка данных включает координаты (x, y) и значение параметра (value). Также в ответе присутствуют диапазоны значений долготы и широты. Схема процесса обработки представлена на рисунке 12.



Рисунок 12 – Алгоритм подготовки данных для визуализации типа «тепловая карта»

3.3.2 Алгоритм проведения статистического анализа

Процесс проведения статистического анализа данных микроклимата территории представлен на рисунке 13. Процесс включает несколько ключевых этапов, направленных на получение объективной и комплексной картины состояния микроклимата. Для описания состояния микроклимата были выбраны методы расчета средних значений, медианы, стандартного отклонения, выявления аномальных значений, а также сравнения полученных данных с установленными экологическими нормативами и стандартами.

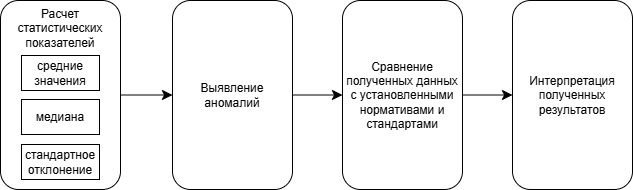


Рисунок 13 – Этапы проведения статистического анализа данных микроклимата

На первом этапе проводится расчет основных статистических показателей, которые позволят получить общее представление о состоянии микроклимата:

* средние значения;
* медиана;
* стандартное отклонение.

Среднее значение позволяет получить общую тенденцию и характерные условия для данного периода. Алгоритм расчета представлен на рисунке 14. В расчет среднего значения входят следующие шаги:

* извлечение значений из DataFrame, который содержит данные по параметру за определенный период;
* суммирование всех значений параметра;
* деление полученной суммы на общее количество наблюдений;
* получение результирующего значения за заданный период.



Рисунок 14 – Алгоритм расчета среднего значения

Медиана является полезным показателем для оценки центральной тенденции, особенно в случаях, когда данные имеют асимметричное распределение или содержат выбросы. Алгоритм расчета представлен на рисунке 15. Расчет состоит из следующих этапов:

* извлечение значений из DataFrame, который содержит данные по параметру за определенный период;
* сортировка значений параметра в порядке возрастания;
* определение середины, если количество наблюдений нечетное, то медиана - это значение, находящееся в середине отсортированного списка, а если количество наблюдений четное, то медиана - это среднее арифметическое двух значений, находящихся посередине;
* получение результирующего значения за заданный период.

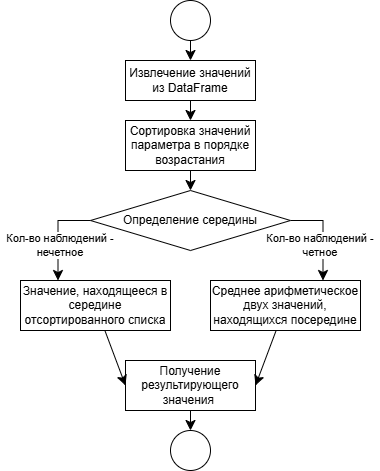


Рисунок 15 – Алгоритм расчета медианы

Стандартное отклонение для каждого параметра показывает степень вариативности данных. Высокое значение стандартного отклонения указывает на значительные колебания параметра, тогда как низкое значение свидетельствует о стабильности условий. Алгоритм расчета представлен на рисунке 16. Расчет происходит в следующем порядке:

* извлечение значений из DataFrame, который содержит данные по параметру за определенный период;
* для каждого наблюдения производится вычисление разницы между его значением и средним значением параметра;
* возведение каждого отклонения в квадрат;
* суммирование всех полученных квадратов отклонений;
* деление суммы квадратов отклонений на количество наблюдений;
* извлечение квадратного корня из полученного значения;
* получение результирующего значения за заданный период.



Рисунок 16 – Алгоритм расчета среднего отклонения

Для более глубокого анализа состояния микроклимата на следующем этапе проводится выявление аномальных значений. Расчет основан на методе z-score. Подробное описание алгоритма представлено на рисунке 17. Экстремальные значения важны для выявления периодов с экстремальными климатическими условиями, которые могут оказать значительное влияние на окружающую среду и человеческую деятельность.

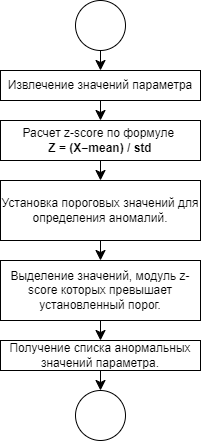


Рисунок 17 – Алгоритм выявление аномальных значений

Далее мы получаем информацию о том, какой процент от общего числа значений составляют аномальные значения, а также список дат, на которых были обнаружены эти аномалии. Это позволяет оценить степень аномальности данных. Алгоритм расчета процентного содержания аномальных значений представлено на рисунке 18. Алгоритм формирования списка дат с аномальными значениями представлен на рисунке 19.

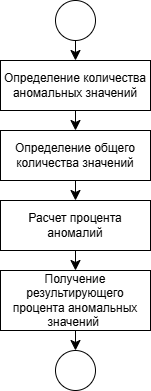


Рисунок 18 – Алгоритм расчета процентного содержания аномальных значений

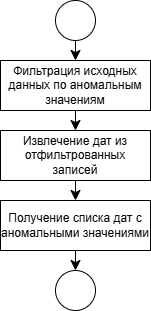


Рисунок 19 – Алгоритм формирования списка дат с аномальными значениями

Таким образом, процесс проведения статистического анализа данных микроклимата позволяет не только получить количественную оценку состояния микроклимата, но и выявить отклонения и экстремальные, что является основой для принятия обоснованных решений и планирования мероприятий по улучшению экологической обстановки.

3.4 Разработка клиентской части

3.4.1 Разработка информационных карточек на главной странице

На главной странице представлены информационные карточки, каждая из которых содержит краткую информацию о графиках, диаграммах и тепловых картах. На каждой карточке также располагается кнопка для перехода на страницу с соответствующим инструментом визуализации. Для реализации карточек был использован компонент фреймворка Quasar [11] q-card. Пример информационных карточек представлен на рисунке 20.

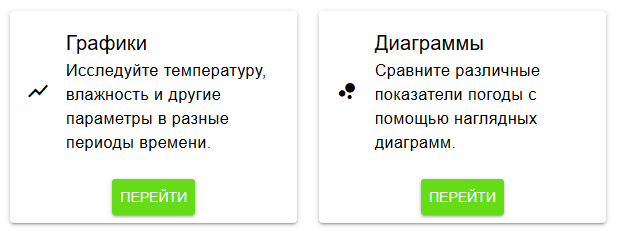


Рисунок 20 – Информационные карточки

3.4.2 Разработка формы загрузки файла

Форма загрузки данных представляет собой интерактивный интерфейс, который позволяет пользователям загружать CSV-файлы с данными о микроклимате и экологии. Форма представлена в виде элемента для выбора файла, реализованная с помощью компонента q-file. При нажатии на кнопку «Выберите файл» открывается диалоговое окно для выбора файла с компьютера. После выбора файла пользователь нажимает на кнопку «Загрузить», после чего отправляется запрос на сервер для загрузки данных из файла в бд. Форма загрузки представлена на рисунке 21.

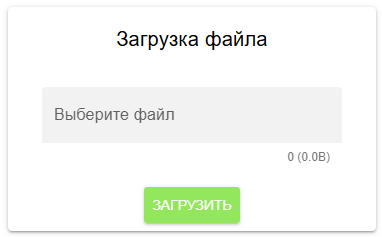


Рисунок 21 – Форма загрузки файла

3.4.3 Разработка секции для выбора параметров визуализации

С помощью компонентов q-date и q-select создается панель с элементами управления, позволяющими пользователю выбирать параметры для визуализации. Элементы управления включают выпадающие списки и выбор диапазона дат. Кнопка «Показать» остается недоступной, пока не будут выбраны все параметры. Пользователь может выбрать параметры, которые хочет увидеть на графиках, диаграммах или тепловых картах. После выбора пользователем параметров и диапазона дат, данные отправляются на сервер для обработки и подготовки к визуализации. После успешного получения данных с сервера визуализации на графиках, диаграммах и тепловых картах автоматически обновляются, отображая новые данные в соответствии с выбранными параметрами и диапазоном дат. Панель для выбора параметров представлена на рисунке 22.



Рисунок 22 – Панель выбора параметров визуализации

3.4.4 Разработка визуализации графика и таблицы с данными

Разработка секции включает создание интерактивного и информативного представления данных пользователю в виде графиков и информации в таблице. В данном случае используется график line, который позволяет визуализировать отношение между двумя переменными (датой и значением). При наведении мыши на точки графика отображается информация о соответствующих значениях. При изменении параметров визуализации или выборе другого диапазона дат данные на графике и в таблице автоматически обновляются. Для реализации визуализации использована библиотека Chart.js [15]. Таблица реализована с помощью компонента q-table. График с таблицей представлен на рисунке 23.

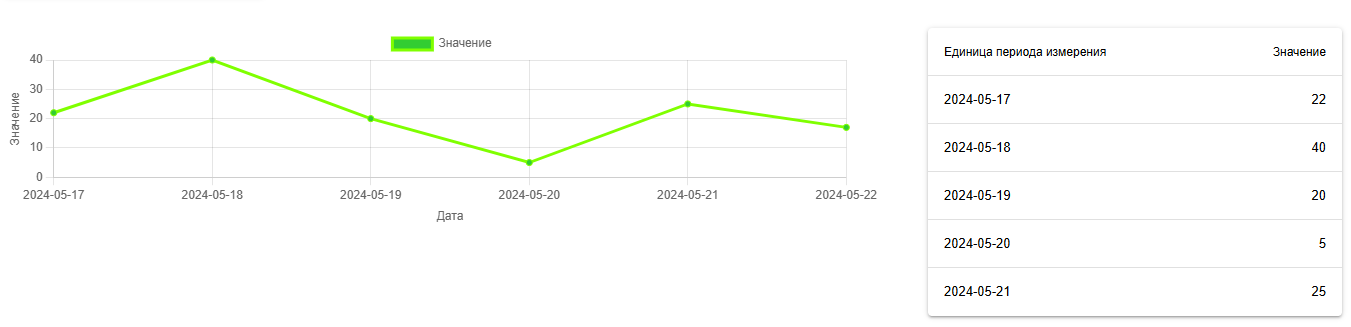


Рисунок 23 – График и таблица с данными

3.4.5 Разработка секции статистического анализа

Среднее значение, медиана, стандартное отклонение и аномальные значения представлены в виде таблицы. Данная секция реализована с помощью компонента q-table, который заполняется данными, полученными с сервера. Также в этой секции представлена текстовая интерпретация результатов по процентному содержанию аномальных значений с выводом списка дат. Секция представлена на рисунке 24.

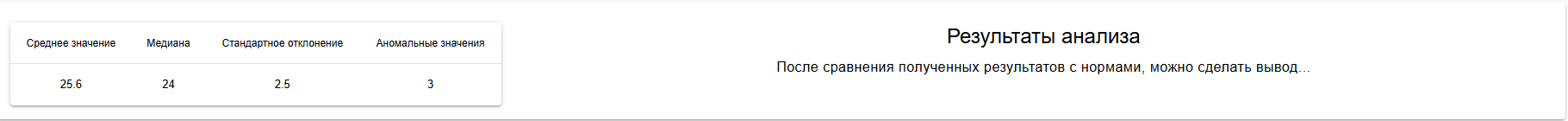


Рисунок 24 – Секция статистического анализа

3.4.6 Разработка визуализации диаграммы и таблицы с данными

Разработка визуализации диаграммы распределения и таблицы с данными реализована аналогично графику, с использованием библиотеки Chart.js [15]. В данном случае используется тип графика scatter. Отличия от графика заключены не только в типе графика, но и в подготовке данных к визуализации. Диаграмма и таблица представлены на рисунке 25.

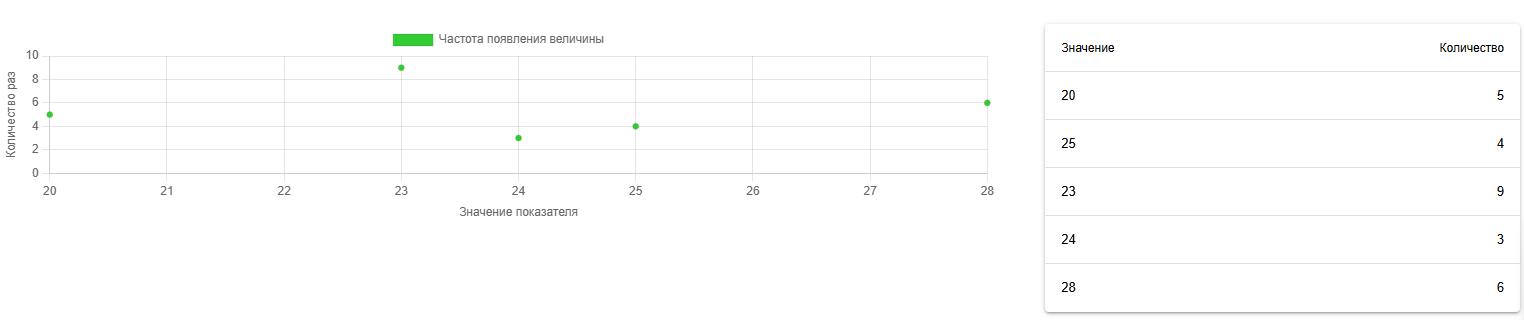


Рисунок 25 – Диаграмма распределения и таблица с данными.

3.4.7 Разработка визуализации тепловой карты

Разработка секции тепловой карты включает в себя создание интерактивного и информативного представления географического распределения данных пользователю. В данном случае используется тепловая карта, которая позволяет визуализировать интенсивность параметра в различных географических точках. Для реализации визуализации использована библиотека d3.js [16]. Таблица реализована с помощью компонента q-table. Тепловая карта с таблицей представлен на рисунке 26.

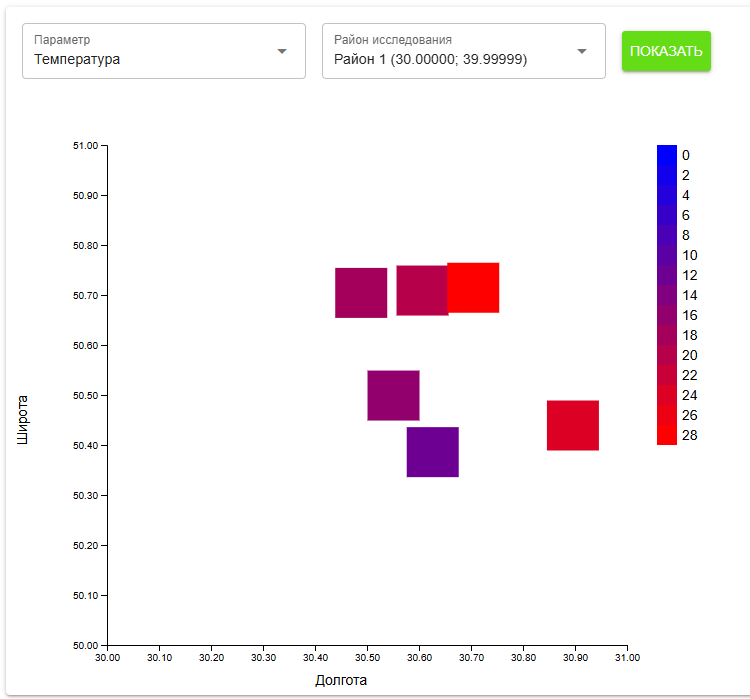


Рисунок 26 – Тепловая карта и таблица с данными

4 Тестирование веб-сервиса

4.1 Описание функционала разработанного веб-сервиса

Веб-сервис спроектирован и реализован для визуализации данных мониторинга микроклимата территории. Веб-сервис имеет главную страницу с описанием функционала, представленную на рисунке 27. Страница имеет карточки с описанием разделов веб-сервиса с кнопками для перехода на соответствующие страницы. Переходы по разделам веб-сервиса также может осуществляться по кнопкам в header страницы.

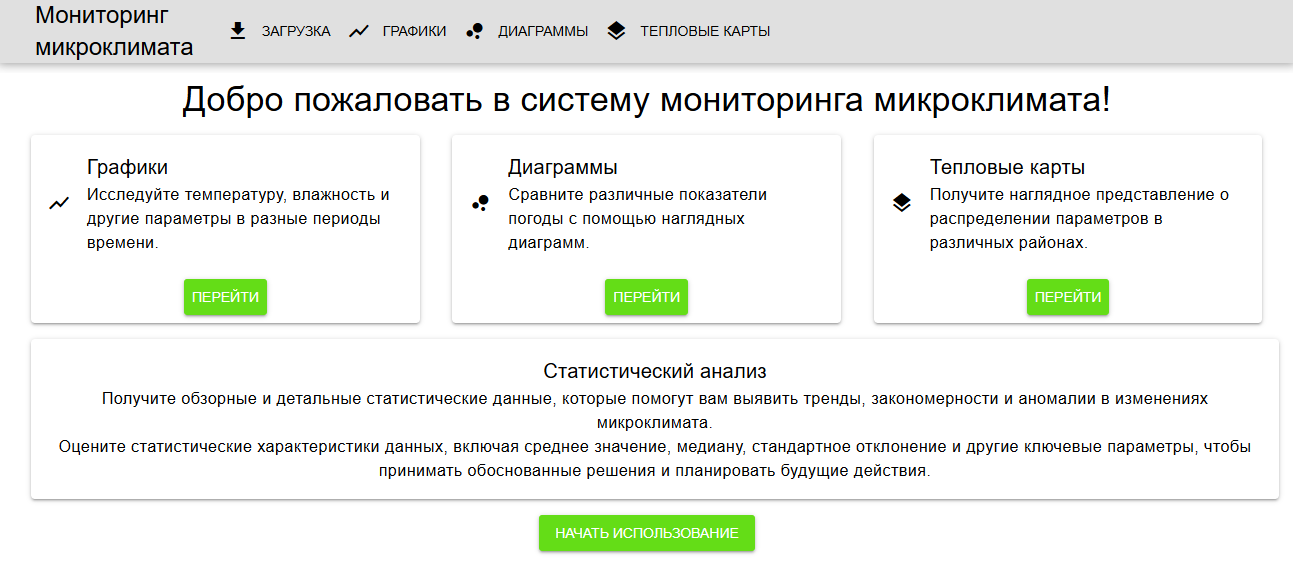


Рисунок 27 – Экранная форма главной страницы

При нажатии на кнопку «Начать использование» пользователь переходит на страницу загрузки данных, представленной на рисунке 28.

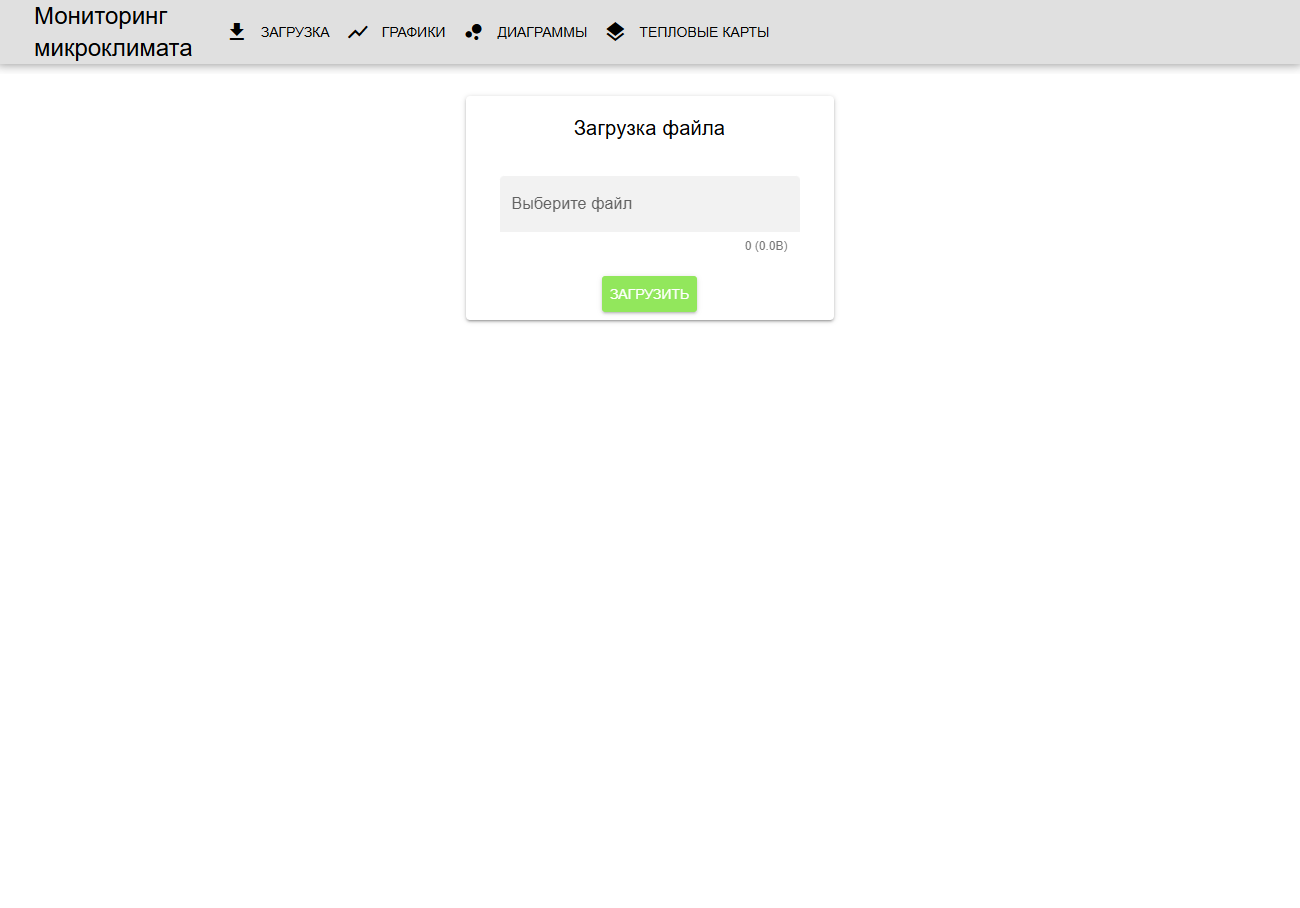
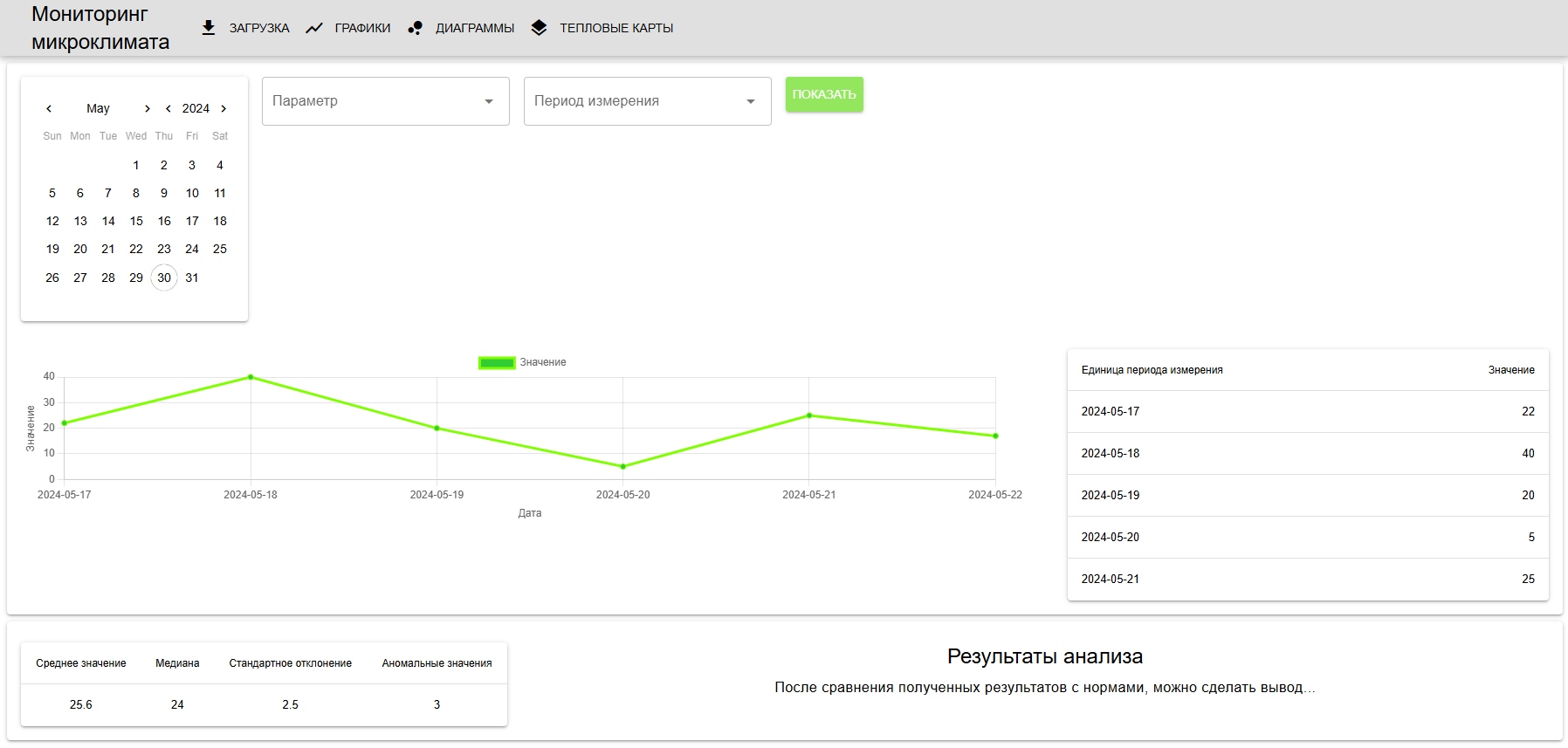


Рисунок 28 – Экранная форма страницы загрузки данных

При нажатии на кнопку «Графики» пользователь переходит на страницу с визуализацией данных в виде графиков, представленной на рисунке 29. На этой странице пользователь может выбрать диапазон дат, параметр и период измерения и сгенерировать график. Также на странице отображаются результаты статистического анализа.

 Рисунок 29 – Экранная форма страницы с графиками

При нажатии на кнопку «Диаграммы» пользователь переходит на страницу с визуализацией данных в виде диаграмм, представленной на рисунке 30. На этой странице пользователь также может выбрать диапазон дат, параметр и период измерения и сгенерировать диаграмму.

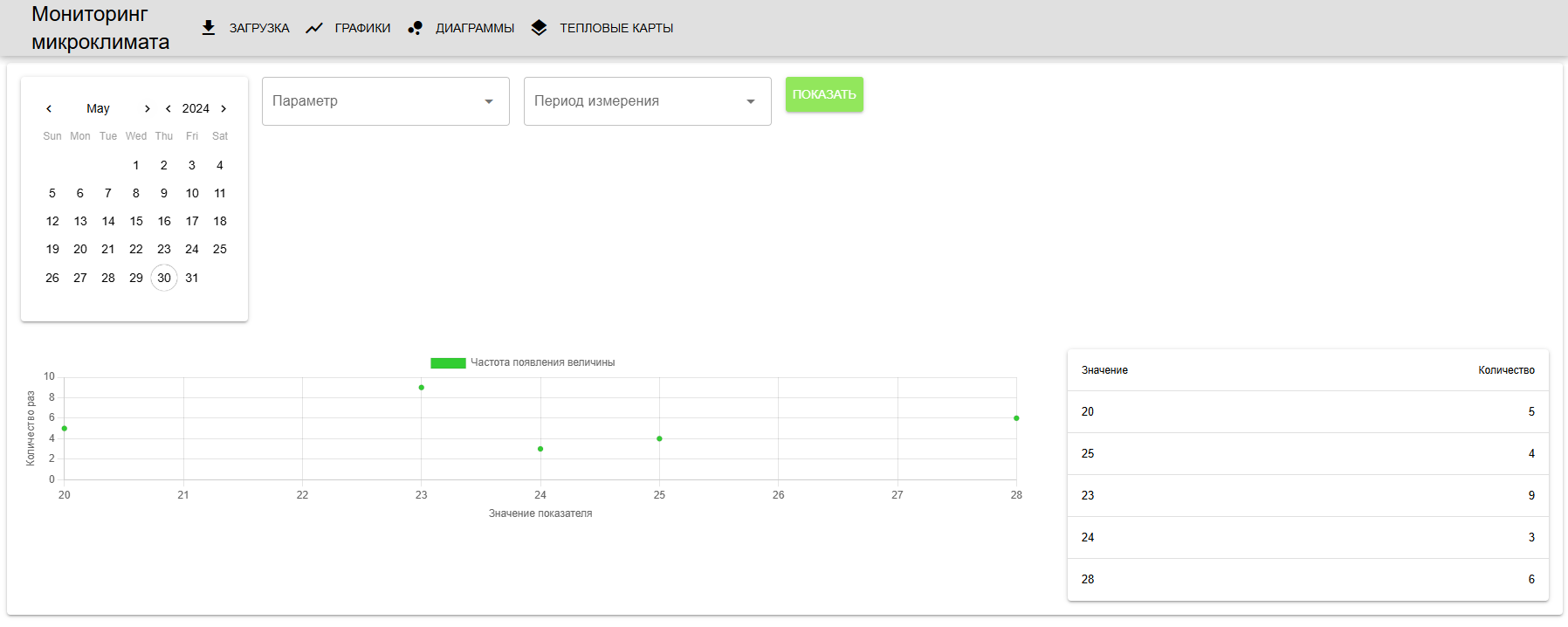


Рисунок 30 – Экранная форма страницы с диаграммами

При нажатии на кнопку «Тепловые карты» пользователь переходит на страницу с визуализацией данных в виде тепловой карты, представленной на рисунке 31. На этой странице пользователь может выбрать параметр и район исследования и сгенерировать карту.

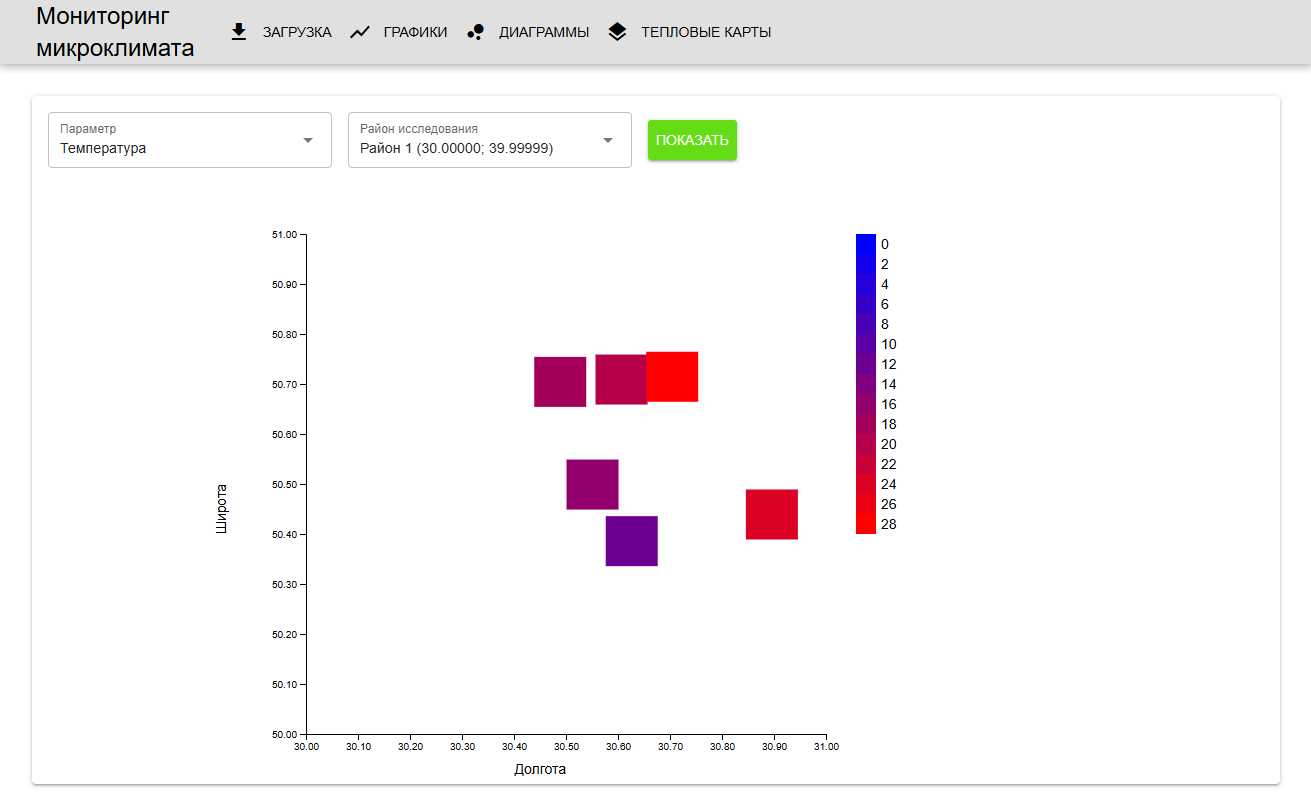


Рисунок 31 – Экранная форма страницы с тепловой картой

4.2 Входные и выходные данные

Входными данными являются:

* данные о температуре, влажности, скорости ветра, давлении и других параметрах, собранные с метеостанций, датчиков и других источников;
* географические координаты местоположения каждой метеостанции или датчика;
* временные метки, указывающие время сбора данных.

Выходными данными являются:

* графики, диаграммы и карты, отображающие изменения параметров микроклимата на городской территории;
* тепловые карты, демонстрирующие распределение температуры, влажности и других показателей;
* аналитические данные, включающие в себя статистические показатели.

4.3 Функциональное тестирование

Для функционального тестирования, необходимо обозначить шаги, которые должна выполнять система. Шаги приведены в таблице 2.

Таблица 2 – Таблица функционального тестирования программы

| № | Шаг | Действие пользователя | Ожидание от программы | Результат |
| --- | --- | --- | --- | --- |
| 1 | Запуск веб-системы | Открытие сайта веб-системы | Открытие главной страницы с описанием функционала веб-сервиса | Выполнено |
| 2 | Навигация | Нажатие кнопки "Загрузка" | Открытие страницы для загрузки данных | Выполнено |
| 3 | Загрузка данных | Выбор файла для загрузки, нажатие на кнопку загрузить | Открытие окна для выбора файла, загрузка данных из файла в БД | Выполнено |
| 4 | Навигация | Нажатие кнопки "Графики" | Открытие страницы с визуализацией в виде графиков | Выполнено |
| 5 | Выбор параметров | Выбор параметра и периода измерения, нажатие кнопки «Показать» | График обновился в соответствии с выбранными параметрами | Выполнено |
| 6 | Интерактивность | Перемещение курсора по графику | Отображение информации о точке на графике | Выполнено |
| 7 | Навигация | Нажатие кнопки "Диаграммы" | Открытие страницы с визуализацией в виде диаграмм | Выполнено |
| 8 | Выбор параметров | Выбор параметра и периода измерения, нажатие кнопки «Показать» | Диаграмма обновилась в соответствии с выбранными параметрами | Выполнено |
| 9 | Интерактивность | Наведение на элемент диаграммы | Отображение информации о выбранном элементе диаграммы | Выполнено |
| 10 | Навигация | Нажатие кнопки "Тепловые карты" | Открытие страницы с визуализацией в виде тепловой карты | Выполнено |
| 11 | Выбор параметров | Выбор параметра и района исследования, нажатие кнопки «Показать» | Тепловая карта обновилась в соответствии с выбранными параметрами | Выполнено |
| 12 | Завершение работы | Закрытие веб-сервиса | Веб-сервис успешно закрывается | Выполнено |

Таким образом, все требуемые функции были реализованы и протестированы, вследствие чего не было выявлено ошибок или несоответствий.

4.4 Тестирование отказоустойчивости

Тестирование отказоустойчивости заключалось в воспроизведении нагрузки на серверную часть разработанного программного комплекса. Нагрузка совершалась за счет одновременной подачи нескольких запросов от нескольких пользователей.

В процессе тестирования отказоустойчивости было проведено 3 теста. Их описание приведено ниже.

Тест 1. Попытка загрузки файла неверного формата:

* при загрузке данных система не дает выбрать файл неверного формата, предотвращая загрузку.

Тест 2. Попытка нажатия кнопки «Загрузить» при отсутствии выбранного файла:

* кнопка «Загрузить» остается недоступной до тех пор, пока не будет выбран файл.

Тест 3. Попытка сгенерировать график, диаграмму или тепловую карту без выбора параметров:

* кнопка «Показать» остается недоступной до тех пор, пока не будут выбраны все параметры.

4.5 Тестирование удобства интерфейса

Юзабилити тестирование прототипа проводилось с 5 студентами ВолгГТУ, после тестирования был составлен список для улучшения интерфейса:

* первый пользователь выявил необходимость видеть на тепловой карте более четкие и понятные цветовые градиенты;
* второй пользователь предложил добавить поясняющие подсказки для кнопок и элементов интерфейса;
* третий пользователь хотел бы видеть более подробные описания функционала на главной странице;
* четвертый пользователь отметил необходимость улучшения визуального оформления;
* пятый пользователь предпочел бы иметь возможность быстро переключаться между различными разделами системы.

После улучшения интерфейса было проведено повторное тестирование со студентами ВолгГТУ, был составлен отчёт по тестированию, отдельные результаты которого представлены далее.

Отчёт по тестированию главного экрана:

* первый пользователь оценил более четкие и понятные цветовые градиенты на тепловой карте;
* второй пользователь получил поясняющие подсказки для всех кнопок и элементов интерфейса;
* третий пользователь увидел более подробные описания функционала на главной странице;
* четвертый пользователь отметил улучшение визуального оформления;
* пятый пользователь отметил, что навигация по разделам системы стала более понятной.

4.6 Тестирование запросов на сервер

Тестирование осуществляется с помощью Postman [24]. Проверяем запрос на добавление файла с данными. Создаем новый POST-запрос с URL-адресом эндпоинта: http://127.0.0.1:8000/upload. Выбираем файл для загрузки, пример представлен на рисунке 32. После отправки запроса Postman должен вернуть успешный статус код, а данные из CSV-файла будут добавлены в базу данных. Результат выполнения запроса представлен на рисунке 33.

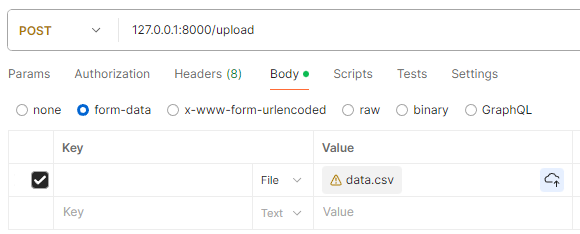


Рисунок 32 – Пример добавления файла

Рисунок 33 – Результат выполнения запроса на загрузку данных

Проверяем запрос на получения данных для генерации графика. Создаем новый GET-запрос с URL-адресом эндпоинта, содержащим параметры визуализации: 127.0.0.1:8000/graphs?parametr=temperature&date\_from=2021-06-01&date\_to=2021-06-30&grouping=day. Тело запроса представлено на рисунке 34. После отправки запроса Postman [24] отобразит ответ от сервера, содержащий массив точек (points), среднее значение (mean), медиану (median), стандартное отклонение (stDeviation) и массив аномальных значений (abnormal). Результат выполнения запроса представлен на рисунке 35.

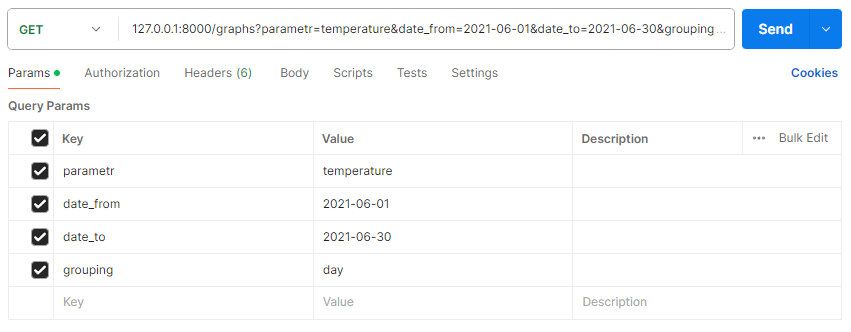


Рисунок 34 – Тело запроса на получение данных для генерации графика

Рисунок 35 – Результат выполнения запроса на получение данных для генерации графика с параметром «температура»

На рисунке 36 и рисунке 37 представлены результаты запросов для получения данных для генерации данных с параметрами «влажность» и «уровень загрязнения» соответственно.

Рисунок 36 – Результат выполнения запроса на получение данных для генерации графика с параметром «влажность»

Рисунок 37 – Результат выполнения запроса на получение данных для генерации графика с параметром «уровень загрязнения»

Проверяем запрос на получения данных для генерации диаграммы распределения. Создаем новый GET-запрос с URL-адресом эндпоинта, содержащим параметры визуализации: 127.0.0.1:8000/diagrams?parametr=temperature&date\_from=2021-06-01&date\_to=2021-06-30. Тело запроса представлено на рисунке 38. После отправки запроса Postman [24] отобразит ответ от сервера, содержащий массив точек (points) для диаграммы. Результат выполнения запроса представлен на рисунке 39.

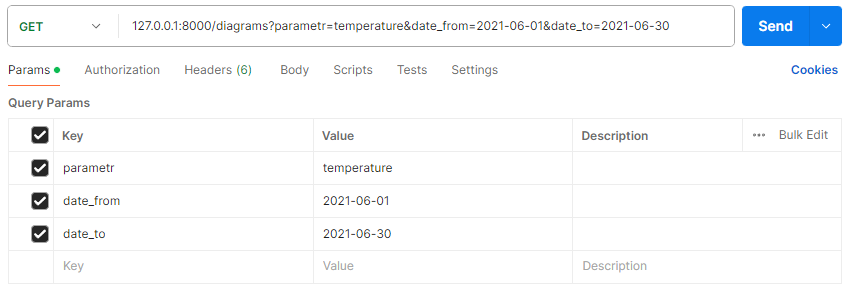


Рисунок 38 – Тело запроса на получение данных для генерации диаграммы

Рисунок 39 – Результат выполнения запроса на получение данных для генерации диаграммы

Проверяем запрос на получения данных для генерации тепловой карты. Создаем новый GET-запрос с URL-адресом эндпоинта, содержащим параметры визуализации: http://localhost:8000/heatmaps?parametr=temperature&lat\_min=9&lat\_max=37.8049&lon\_min=-122.4194&lon\_max=-122.3494. Тело запроса представлено на рисунке 40. После отправки запроса Postman [24] отобразит ответ от сервера, содержащий массив точек (points) для тепловой карты, а также значения доменов по осям X и Y (xDomain и yDomain). Результат выполнения запроса представлен на рисунке 41.

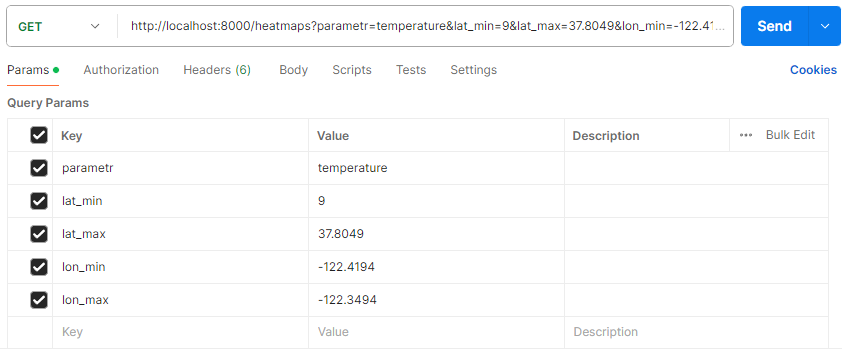


Рисунок 40 – Тело запроса на получение данных для генерации диаграммы

Рисунок 41 – Результат выполнения запроса на получение данных для генерации диаграммы

Заключение

В результате выполненной работы была разработана и реализована система для обработки и визуализации данных мониторинга микроклимата территории. Этот веб-сервис предоставляет функционал по аналитической обработке данных о температуре, влажности и уровне загрязнения воздуха с целью создания наглядных визуализаций.

Область применения разработанной системы включает в себя комплексные программные решения в области анализа информации о микроклимате из различных источников данных. Система также способствует принятию решений в области мониторинга и управления микроклиматом, что является важным элементом в устойчивом развитии городской среды.

Основными направлениями дальнейшего развития системы являются расширение функционала по увеличению количества учитываемых параметров мониторинга, улучшение алгоритмов анализа данных для более точных прогнозов и развитие модулей визуализации для предоставления пользователю максимально полной и информативной картины состояния микроклимата. Кроме того, важным аспектом будет интеграция с другими системами мониторинга и управления окружающей средой для создания единой информационной платформы по управлению городским микроклиматом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Подведены итоги жилищного строительства в 2023 году // Минстрой России. – 2024. – URL: https://minstroyrf.gov.ru/press/podvedeny-itogi-zhilishchnogo-stroitelstva-v-2023-godu/ (дата обращения: 02.03.2024).

2. Разработка автоматизированной системы экологического мониторинга с применением технологий IoT / Р. Б. Салихов, А. А. Гаскарова, К. В. Важдаев, А. Б. Аллабердин  // Науки о земле. – 2023. – №2. – С. 60-72.

3. Епринцев, С. А. Геоинформационное картографирование урбанизированных территорий как механизм пространственной оценки социально-экологических факторов / С. А. Епринцев, С. В. Шекоян //  Геополитика и экогеодинамика регионов. – 2019. – Т. 5, №3. – С. 109-115.

4. Map Viewer // ArcGIS : официальный сайт. – URL: https://www.arcgis.com/index.html (дата обращения: 02.03.2024).

5. Smart Citizen // Smart Citizen : официальный сайт. – URL: https://smartcitizen.me/kits/ (дата обращения: 02.03.2024).

6. EcoAtlas // EcoAtlas : официальный сайт. – URL: https://www.ecoatlas.org/ (дата обращения: 02.03.2024).

7. Моделирование бизнес-процессов, автоматический перевод «диаграмма-текст» и СН-1 нотация // HABR. – 2020. – URL: https://habr.com/ru/articles/500104/ (дата обращения: 05.04.2024).

8. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // HABR. – 2021. – URL: https://habr.com/ru/articles/566218/ (дата обращения: 08.04.2024).

9. Клиент серверная архитектура в картинках // HABR. – 2020. – URL: https://habr.com/ru/articles/495698/ (дата обращения: 12.04.2024).

10. Введение | Vue.js : [документация Vue.js] // Vue.js : официальный сайт. – URL: https://v3.ru.vuejs.org/ru/guide/introduction.html (дата обращения: 01.02.2024).

11. Quasar Docs : [документация Quasar Framework] // Quasar : официальный сайт. – URL: https://quasar.dev/docs (дата обращения: 08.02.2024).

12. FastAPI : [документация FastAPI] // FastAPI : официальный сайт. – URL: https://fastapi.tiangolo.com/ (дата обращения: 20.04.2024).

13. PostgreSQL: Documentation : [документация PostgreSQL] // PostgreSQL : официальный сайт. – URL: https://www.postgresql.org/docs/ (дата обращения: 01.02.2024).

14. 3.12.3 Documentation : [документация Python] // Python : официальный сайт. – URL: https://docs.python.org/3/ (дата обращения: 20.04.2024).

15. Chart.js | Chart.js : [документация Chart.js] // Chart.js : официальный сайт. – URL: https://www.chartjs.org/docs/latest/ (дата обращения: 10.02.2024).

16. Getting started | D3 by Observable : [документация d3.js] // d3.js : официальный сайт. – URL: https://d3js.org/getting-started (дата обращения: 10.02.2024).

17. pandas documentation – pandas : [документация pandas] // pandas : официальный сайт. – URL: https://pandas.pydata.org/docs/ (дата обращения: 20.04.2024).

18. NumPy documentation – NumPy : [документация NumPy] // NumPy : официальный сайт. – URL: https://numpy.org/devdocs/ (дата обращения: 20.04.2024).

19. csv – CSV File Reading and Writing : [документация csv] // Python : официальный сайт. – URL: https://docs.python.org/3/library/csv.html (дата обращения: 20.04.2024).

20. SQLAlchemy Documentation – SQLAlchemy 2.0 Documentation : [документация SQLAlchemy] // SQLAlchemy : официальный сайт. – URL: https://docs.sqlalchemy.org/en/20/ (дата обращения: 20.04.2024).

21. Git - Reference : [документация Git] // Git : официальный сайт. – URL: https://git-scm.com/docs (дата обращения: 01.02.2024).

22. Documentation for Visual Studio Code : [документация Visual Studio Code] // Visual Studio Code : официальный сайт. – URL: https://code.visualstudio.com/docs (дата обращения: 01.02.2024).

23. Getting started | PyCharm Documentation : [документация PyCharm] // JetBrains : официальный сайт. – URL: https://www.jetbrains.com/help/pycharm/getting-started.html (дата обращения: 15.04.2024).

24. Postman documentation overview : [документация Postman] // Postman : официальный сайт. – URL: https://learning.postman.com/docs/introduction/overview/ (дата обращения: 25.04.2024).

Приложение А  
Техническое задание