**СОДЕРЖАНИЕ**

Перечень сокращений и обозначений 3

Введение 4

1. Анализ и разработка требований 6
   1. Назначение и область применения 6
   2. Постановка задачи 7
   3. Выбор состава технических средств 9
2. Проектирование программного обеспечения 11
   1. Проектирование базы данных 11
3. Разработка программного обеспечения 14
   1. Разработка и интеграция программных модулей 14
   2. Реализация защиты данных 17
   3. Тестирование программного обеспечения 19
4. Инструкция по эксплуатации программного обеспечения 22
   1. Конфигурация программного обеспечения 22
   2. Установка программного обеспечения 23
   3. Руководство по эксплуатации 23
5. Определение затрат на разработку 34
   1. Определение затрат на разработку 34
   2. Анализ экономического обоснования и рисков проекта 46
6. Охрана труда и техника безопасности при работе на ПК 48
   1. Общие требования охраны труда 48
   2. Требования охраны труда перед началом работы 48
   3. Требования охраны труда во время работы 49
   4. Требования охраны труда по окончании работы 50
   5. Требования охраны труда в аварийных ситуациях 50

Заключение 51

Список использованных источников 52

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

В настоящем дипломном проекте применяются следующие сокращения и обозначения:

АСОИЗА – автоматизированная система обработки информации загрязнения атмосферы

БД – база данных

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

УГМС – управление по гидрометеорологии и мониторингу окружающей среды

ФГБУ – федеральное государственное бюджетное учреждение

ЭВМ – электронная вычислительная машина

API – интерфейс прикладного программирования

ASP – платформа разработки веб-приложений

CQRS – стиль архитектуры, в котором операции чтения отделены от операций записи

HTTP – протокол передачи гипертекста

ORM – объектно-реляционное отображение

SHA – безопасный алгоритм хеширования

UML – унифицированный язык моделирования

URL – унифицированный указатель ресурса

**ВВЕДЕНИЕ**

Актуальность дипломного проектирования заключается в том, что в современном мире проблема загрязнения воздуха становится все более актуальной. Индустриализация, рост числа автомобилей и другие антропогенные факторы приводят к ухудшению качества воздуха, что негативно сказывается на здоровье людей и состоянии окружающей среды. Загрязнение воздуха вызывает множество респираторных заболеваний, таких как астма и хронический бронхит, а также сердечно-сосудистые заболевания. Кроме того, загрязненный воздух способствует изменению климата, что приводит к более частым и интенсивным природным катаклизмам, таким как ураганы, наводнения и засухи, что оказывает негативное влияние на сельское хозяйство и водные ресурсы, создавая дополнительные экологические и экономические проблемы.

В связи с этим возникает необходимость в создании систем, которые могут автоматически собирать и анализировать данные о состоянии воздуха, чтобы своевременно выявлять и устранять источники загрязнения. Создание и внедрение автоматизированной системы мониторинга качества воздуха является важным шагом на пути к улучшению экологической ситуации и повышению качества жизни.

Целью дипломного проектирования является разработка системы автоматического сбора данных «Воздух». Эта система предназначена для мониторинга показателей загрязнения воздуха, а также таких измерений, как температура, давление и общие показатели состояния атмосферы. Система должна собирать данные с программного продукта «АСОИЗА+», обрабатывать их и предоставлять пользователю в удобной форме для анализа.

Для достижения поставленной цели требуется решить следующие задачи:

* собрать и проанализировать функциональные и эксплуатационные требования к ПО,
* описать работу программного модуля с помощью UML,
* выбрать состав программных и технических средств для разрабатываемого ПО,
* выделить объекты и атрибуты БД в соответствии с заданием,
* спроектировать БД,
* разработать БД,
* создать запросы к БД,
* спроектировать архитектуру ПО,
* спроектировать интерфейс пользователя,
* реализовать интерфейс пользователя,
* разработать программные модули системы автоматического сбора данных «Воздух»,
* выполнить тестирование и отладку разработанного ПО,
* разработать справочную документацию,
* описать процесс установки разработанного ПО,
* обеспечить организацию защиты ПО компьютерных систем программными средствами.

**1 Анализ и разработка требований**

**1.1 Назначение и область применения**

ФГБУ «Северное УГМС» является некоммерческой организацией, созданной для обеспечения потребностей государства, юридических и физических лиц в гидрометеорологической, гелиогеофизической информации, в информации о состоянии окружающей среды, ее загрязнении, в том числе экстренной информацией на территориях Архангельской и Вологодской областей, Республики Коми, Ненецкого автономного округа, Ямальского района Ямало-Ненецкого автономного округа, городского поселения Диксон и сельского поселения Хатанга, части территории Республики Карелия, части территории Мурманской области, Северной части Таймырского (Долгано-Ненецкого) муниципального района Красноярского края, акватории Белого и Карского морей, юго-востока Баренцева моря, запада моря Лаптевых.

Разрабатываемое ПО будет внедрено в деятельность ФГБУ «Северное УГМС», где его использование охватит различные аспекты мониторинга окружающей среды. Основная область применения ПО включает в себя автоматический сбор данных с программного продукта «АСОИЗА+», их обработку и визуализацию для дальнейшего анализа специалистами в лаборатории загрязнения атмосферного воздуха.

Система «Воздух» обеспечит высокую степень автоматизации в процессе сбора данных, что позволит ФГБУ «Северное УГМС» оперативно реагировать на изменения в окружающей среде и предоставлять актуальную информацию о состоянии воздуха. Это, в свою очередь, повысит точность прогнозирования экологических рисков, улучшит координацию действий по охране окружающей среды и обеспечит более эффективное использование ресурсов для поддержания экологической безопасности.

**1.2 Постановка задачи**

В ходе дипломного проектирования необходимо разработать систему автоматического сбора данных «Воздух», предназначенную для автоматического сбора данных с программного продукта «АСОИЗА+» в режиме реального времени с последующим хранением собранной информации в течение трех лет. Интерфейс приложения необходимо реализовать в виде веб-интерфейса.

В системе должны быть реализованы следующие задачи автоматизации:

* сбор информации с «АСОИЗА+» и ее сохранение в системе,
* автоматическое удаление неактуальной информации из системы (старше трех лет),
* вывод собранных данных для выбранной станции в виде таблицы с возможностью фильтрации по дате добавления,
* добавление новых пользователей в систему администратором,
* ограничение доступа к данным станций и показателям измерений индивидуально для каждого пользователя,
* анализ собранных данных и предупреждение пользователей при превышении нормы измерений,
* автоматическое прекращение сбора данных со станций, которые не отправляли данные более недели,
* автоматическое начало сбора данных со станций, которые возобновили отправку данных,
* автоматический поиск новых станций и последующее добавление информации о них в систему.

На рисунке 1 представлена диаграмма вариантов использования [20] ПО администратором, пользователем и неавторизованным пользователем.

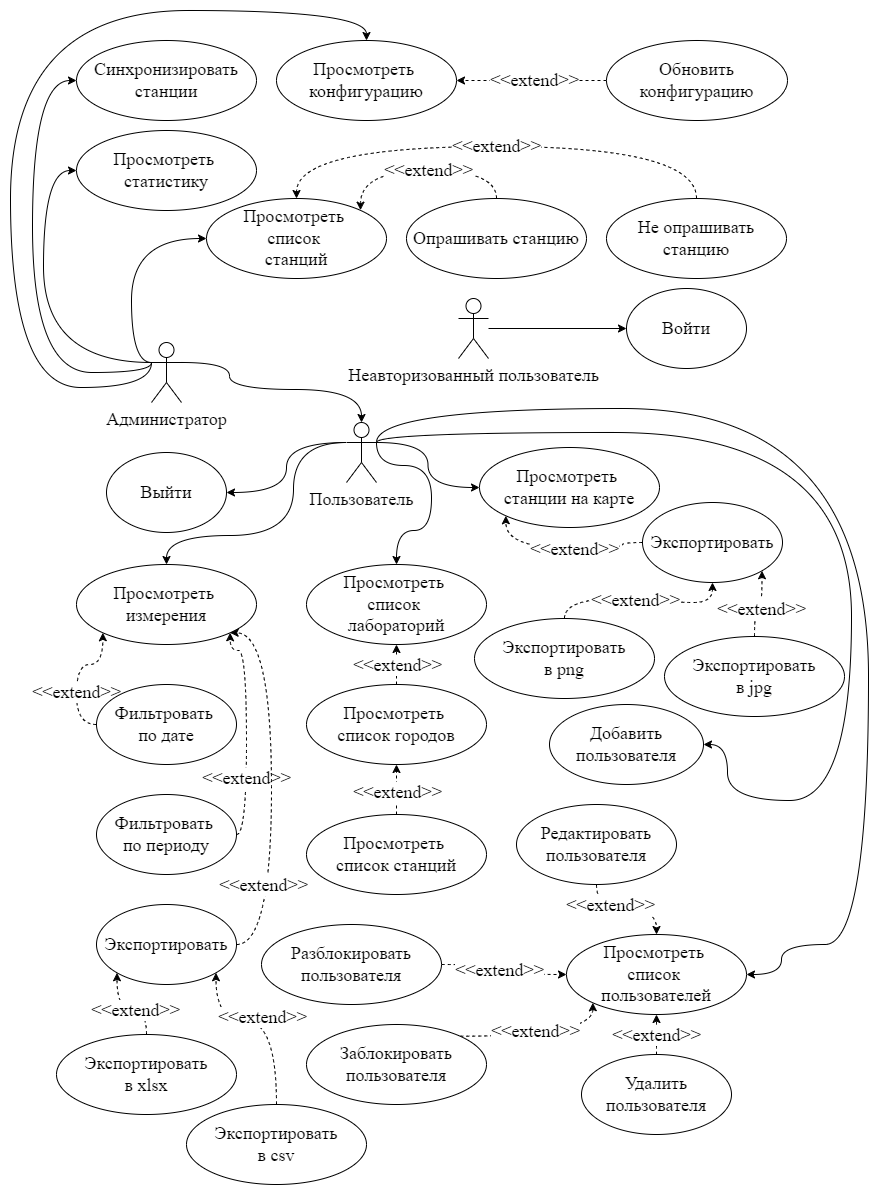


Рисунок 1 – Диаграмма вариантов использования

**1.3 Выбор состава технических средств**

В процессе проектирования системы автоматического сбора данных «Воздух» выбран технологический стек, который включает в себя язык программирования C# [14] для разработки серверной части приложения. Этот выбор обусловлен высокой производительностью и надежностью, которые необходимы для выполнения серверных операций. В процессе разработки программного модуля на платформе .NET будут применены следующие передовые технологии и современные библиотеки:

* «ASP.NET Core» [9] для построения масштабируемого API с использованием языка программирования C#,
* «Entity Framework Core» [16] для удобного взаимодействия с БД, предоставляет ORM-подход к работе с данными,
* «AutoMapper» для автоматического преобразования между объектами различных типов,
* «MediatR» для реализации CQRS-архитектуры, обеспечивая разделение команд и запросов для обеспечения гибкости и расширяемости приложения,
* «FluentValidation» для гибкой валидации входящих данных,
* «Autofac» для реализации механизма внедрения зависимостей, обеспечивая управление зависимостями в приложении.

В качестве СУБД выбрана PostgreSQL [15], известная своей масштабируемостью и надежностью, а также поддержкой сложных запросов и транзакций, что обеспечивает высокую производительность и стабильность работы всей системы.

Для разработки клиентской части приложения выбран язык программирования TypeScript в связке с библиотекой React [19] и библиотекой Ant Design, обеспечивающей широкий набор готовых компонентов.

Для визуализации данных выбрана библиотека Recharts, которая предоставляет мощные инструменты для создания графиков и диаграмм. А для работы с картами и географическими данными выбрана библиотека OpenLayers.

Для эффективного управления состоянием приложения будут использованы библиотеки Redux, Redux Toolkit и redux-persist. Эти библиотеки гарантируют централизованное хранение данных, обеспечивая их доступность и согласованность на всех уровнях приложения.

Выбранный технологический стек позволит эффективно реализовать функциональные возможности системы «Воздух» и обеспечить высокое качество конечного продукта.

**2 Проектирование программного обеспечения**

**2.1 Проектирование базы данных**

В связи с поставленной задачей требуется разработать БД [8] для хранения данных о пользователях, лабораториях, городах, станциях, измерениях и других сущностях. Разработка БД представляет собой важный этап проектирования, направленный на эффективное хранение и управление информацией. Создание физической модели позволяет четко определить структуру данных и взаимосвязи между ними. Проектирование БД предоставляет надежную основу для дальнейшей реализации ПО, обеспечивая надежность и эффективное функционирование приложения.

В рамках процесса проектирования БД [6] разработана физическая модель, которая отображает основные сущности и соответствующие атрибуты, необходимые для реализации функциональности, а именно:

* сущность «Измерения» содержит информацию о каждом анализе воздуха, отправленным станцией, включая идентификатор измерения, дату и время, идентификатор станции, название станции, номер станции, температуру, давление, направление ветра, скорость ветра, влажность, упругость водяного пара, атмосферные явления и показатели загрязнений,
* сущность «Станции» хранит данные о станциях, включая идентификатор станции, ее название, адрес, идентификатор города, идентификатор местоположения и собираются ли данные со станции,
* сущность «Города» представляет информацию о городах, включая идентификатор города, его название, идентификатор лаборатории и идентификатор местоположения,
* сущность «Роли» содержит информацию о ролях пользователей, включая идентификатор роли и ее название,
* сущность «Лаборатории» содержит информацию о лабораториях, включая идентификатор лаборатории, ее название и идентификатор местоположения,
* сущность «Качества» хранит данные об атрибутах для обозначения достоверности измерений, включая идентификатор, его название, описание, цвет для выделения и приоритет,
* сущность «Пользователи» содержит информацию о пользователях, включая идентификатор пользователя, логин, хеш пароля и идентификатор роли,
* сущность «Колонки» содержит информацию о доступных для просмотра показателях измерений, включая идентификатор показателя, ее название, сокращенное название, формулу для расчета показателя и код загрязнения,
* сущность «Сессии» содержит информацию о сессиях пользователей, включая идентификатор сессии, идентификатор пользователя, токен, дату создания и дату продления сессии,
* сущность «Конфигурации» содержит информацию о конфигурации приложения, включая идентификатор конфигурации, логин от «АСОИЗА+», пароль от «АСОИЗА+», интервал сбора данных (в минутах), интервал удаления устаревших данных (в днях), интервал удаления измерений (в месяцах) и интервал отключения станций (в днях),
* сущность «Местоположения» содержит информацию о местоположении лабораторий, городов и станций, включая идентификатор местоположения, ширину и долготу,
* сущность «Статистика сбора данных» содержит информацию о каждом сборе данных со станций, включая идентификатор статистики, дату и время сбора, количество опрошенных станций, длительность сбора и количество собранных измерений.

Разработанная физическая модель БД представлена на рисунке 2.

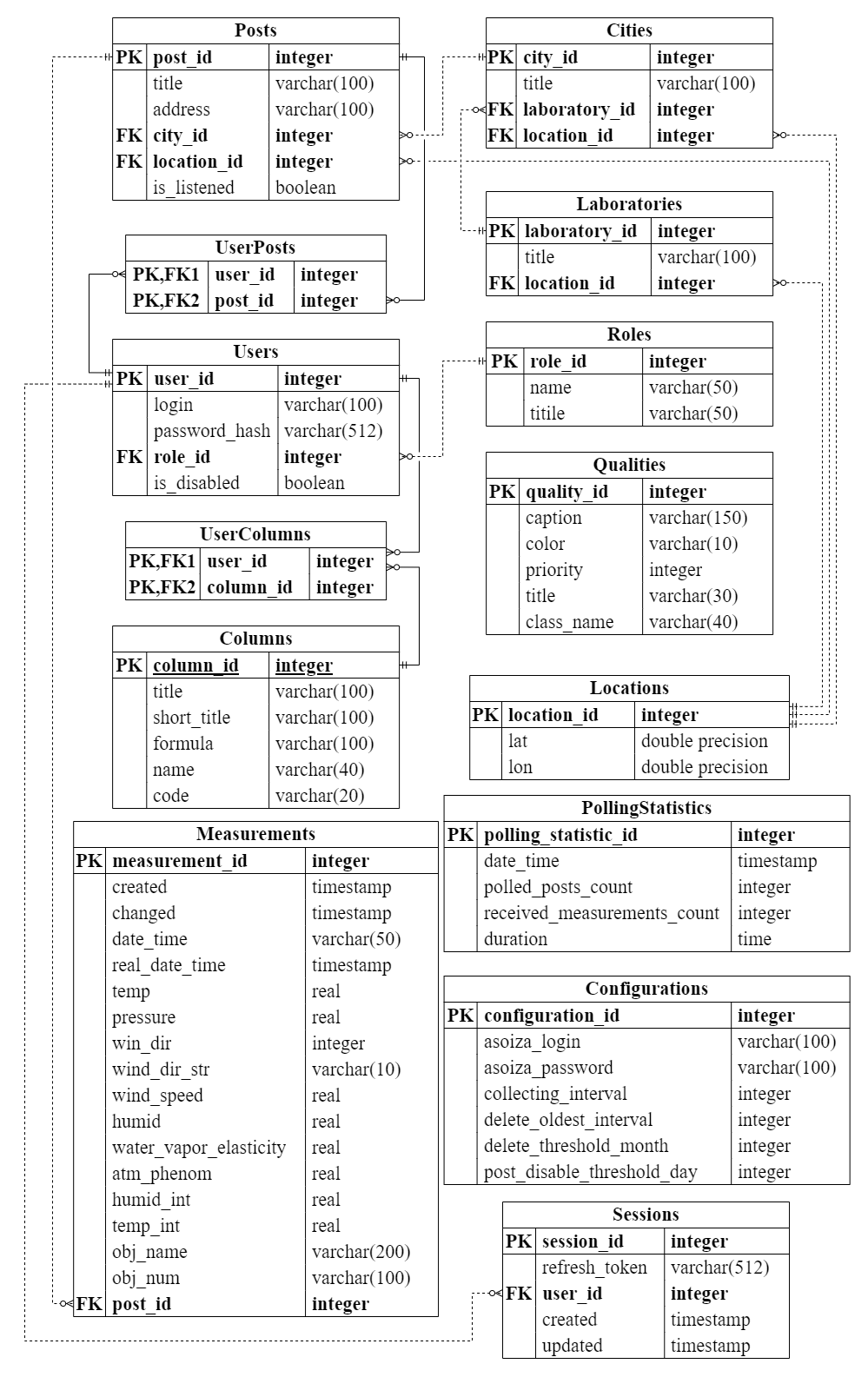


Рисунок 2 – Физическая модель БД

**3 Разработка программного обеспечения**

**3.1 Разработка и интеграция программных модулей**

В процессе разработки системы автоматического сбора данных «Воздух» выбрана трехзвенная архитектура [12] (рисунок 3). Выбор такой архитектуры обусловлен тем, что она обеспечивает четкое разделение логики представления, бизнес-логики и доступа к данным. Также для балансировки нагрузки используется proxy сервер на Nginx [7]. Также Nginx обеспечивает сжатие Gzip, что уменьшает объем передаваемых данных и ускоряет работу приложения.

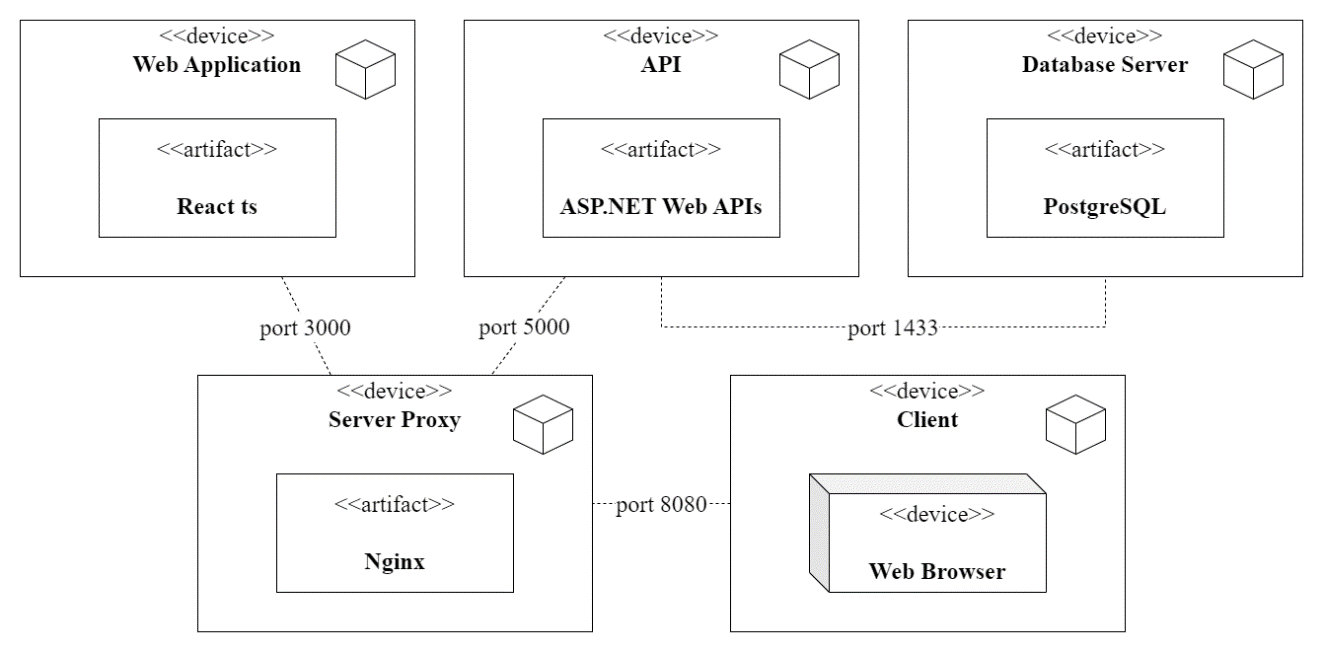


Рисунок 3 – Диаграмма развертывания

В процессе разработки возникла необходимость интеграции с программным продуктом «АСОИЗА+». Для этого разработан класс «AsoizaClient», который позволяет взаимодействовать с «АСОИЗА+» посредством HTTP-запросов. Фрагмент кода разработанного класса представлен листингом 1.

Листинг 1 – Фрагмент кода класса «AsoizaClient»

|  |
| --- |
| public sealed class AsoisaClient(  IWindDirectCalculator directCalculator,  IConfigurationRepository configurationRepository,  IMapper mapper)  : IAsoisaClient  {  private const string BaseUrl = "https://asoiza.voeikovmgo.ru";  private readonly SemaphoreSlim \_semaphore = new(1, 1);  // идентфикатор сессии  private string? \_sessionId;  /// <summary>  /// Получение списка измерений для станции  /// по ее идентификатору  /// </summary>  /// <param name="postId">Идентифкатор станции</param>  /// <param name="count">Количество измерений</param>  /// <param name="skip">Количество пропускаемых измерений</param>  public async Task<Measurement[]> GetMeasurementListAsync(int postId, int count, int skip)  {  // Сегодняшяя дата в utc и временем 23:59:59  var today = DateTime.Now  .ToUniversalTime()  .Date  .AddDays(1)  .AddSeconds(-1)  .ToString("yyyy-MM-dd HH:mm:ss");  // Выполнение запроса с повторной авторизацией в случае необходимости  var response = await BaseRequestWithReAuth(() =>  string.Concat(BaseUrl,  $"/data/MeasurementsStore.php?task=read&mtype=auto&period=month&end={today}&titletype=long&pdktype=sanpin21&presstype=gpa&object={postId}&page=1&start={skip}&limit={count}&sort=%5B%7B%22property%22%3A%22date\_time%22%2C%22direction%22%3A%22DESC%22%7D%2C%7B%22property%22%3A%22obj\_name%22%2C%22direction%22%3A%22ASC%22%7D%5D")  .WithCookie("PHPSESSID", \_sessionId)  .GetAsync());  // Получение строки JSON из ответа  var json = await response.GetStringAsync();  // Проверка, что JSON не null  if (json is null)  {  throw new Exception($"Json for postId: {postId} is null");  }  // Замена ',' на '.' для корректного преобразования значений из JSON  var regex = new Regex("\"v\_\\d{6,}\":\"\\d{1,},\\d{1,}\"");  json = regex.Replace(json, m => m.Value.Replace(",", "."));  // Десериализация JSON-строки в объект AsoizaPostMeasurementsResponse  var result = JsonConvert.DeserializeObject<AsoizaPostMeasurementsResponse>(json);  if (result is null)  {  throw new Exception($"Measurements for postId: {postId} is null");  }  if (result.Data is null)  {  return [];  }  // Обработка каждого элемента данных  foreach (var viewModel in result.Data)  {  viewModel.PostId = postId;  // Расчет направления ветра и преобразование в строку  if (viewModel.WindDir is not null)  {  var winDir = viewModel.WindDir.Value;  viewModel.WindDirStr = $"{directCalculator.WindDirToRmb16(winDir)} ({winDir})";  }  // Расчет времени в UTC, для которого была создана запись  if (viewModel.DateTime is not null)  {  const string format = "yyyy-MM-dd HH:mm:ss";  viewModel.RealDateTime = DateTime.ParseExact(viewModel.DateTime, format, CultureInfo.InvariantCulture)  .ToUniversalTime();  }  }  // Преобразование данных и фильтрация по измененным элементам  return mapper.Map<Measurement[]>(result.Data)  .Where(e => e.Changed is not null)  .ToArray();  }  } |

**3.2 Реализация защиты данных**

В рамках современных требований к информационной безопасности [17] при разработке особое внимание уделялось защите данных и предотвращению несанкционированного доступа. Для этого используется метод авторизации с использованием HTTP-only cookie, который обеспечивает дополнительный уровень безопасности, предотвращая доступ к cookie через клиентские скрипты. Это помогает защитить данные пользователей от различных атак и обеспечивает более надежную аутентификацию. Настройка cookie файлов в серверной части приложения представлена листингом 2.

Листинг 2 – Фрагмент кода настройки cookie файлов

|  |
| --- |
| .AddHttpContextAccessor()  .AddAuthentication("MeasurementsCookieAuth")  .AddCookie("MeasurementsCookieAuth", options =>  {  // Настройка срока действия cookie  options.ExpireTimeSpan = TimeSpan.FromHours(1);  options.SlidingExpiration = true;  options.Cookie.Name = "MeasurementsCookieAuth";  // Включение http only cookie  options.Cookie.HttpOnly = true;  // Включение строгого режима cookie  options.Cookie.SameSite = SameSiteMode.Strict;  options.Events = new CookieAuthenticationEvents  {  // Настройка события при истечении срока действия сессии  OnRedirectToLogin = redirectContext =>  {  // Проверка что это не метод обновления сессии  if (!redirectContext.Request.Path  .ToString().Contains("refresh"))  {  // Установка статуса не авторизован  redirectContext.Response.StatusCode = StatusCodes.Status401Unauthorized;  }  return Task.CompletedTask;  }  };  }); |

Все пароли на стороне сервера шифруются и хранятся в БД в зашифрованном виде. Для криптографического хеширования паролей используется алгоритм SHA-384, который позволяет преобразовать исходный пароль в уникальный хеш [18]. Для повышения устойчивости к атакам методом подбора и словарным атакам, к паролям применяется соль – случайно сгенерированная строка, которая добавляется к паролю перед хешированием. Реализация хеширования паролей осуществляется с помощью библиотеки BCrypt.NET, которая позволяет не только эффективно хешировать пароли, но и осуществлять проверку соответствия введенного пользователем пароля хранимому хешу, что является важным элементом процесса аутентификации. Код класса для шифрования паролей представлен листингом 3.

Листинг 3 – Код класса «PasswordHasher»

|  |
| --- |
| public sealed class PasswordHasher : IPasswordHasher  {  /// <summary>  /// Шифрование пароля  /// </summary>  /// <param name="password">пароль</param>  public string Hash(string password)  {  return BCrypt.Net.BCrypt.EnhancedHashPassword(password, HashType.SHA512);  }  /// <summary>  /// Сверить пароль  /// </summary>  /// <param name="password">пароль</param>  /// <param name="hash">хеш пароля</param>  public bool Verify(string password, string hash)  {  return BCrypt.Net.BCrypt.EnhancedVerify(password, hash, HashType.SHA512);  }  } |

Все перечисленное обеспечивает высокий уровень безопасности и конфиденциальности пользовательской информации, минимизируя риски несанкционированного доступа и утечки данных.

**3.3 Тестирование программного обеспечения**

Для проверки правильности работы приложения необходимо провести тестирование методом «Черного ящика». В таблице 1 представлен набор тестов для формы «Авторизация».

Таблица 1 – Набор тестов формы «Авторизация»

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Полученный результат |
| Нажатие на кнопку «Войти» при заполненных полях «Логин» и «Пароль» данными «Admin» и «Admin» соответственно | Перенаправление пользователя со страницы «Авторизация» на страницу «Измерения» | Совпал с ожиданием |
| Нажатие на кнопку «Войти» при заполненном поле «Логин» данными «Admin» и незаполненном поле «Пароль» | Отображение всплывающего сообщения у поля «Пароль» с текстом «Заполните это поле.» | Совпал с ожиданием |
| Нажатие на кнопку «Войти» при незаполненных полях «Логин» и «Пароль» | Отображение всплывающего сообщения у поля «Логин» с текстом «Заполните это поле.» | Совпал с ожиданием |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Полученный результат |
| Нажатие на кнопку «Войти» при заполненных полях «Логин» и «Пароль» данными «Admin» и «NeAdmin» соответственно | Отображение всплывающего сообщения в верху страницы с текстом «Неверный логин или пароль.» | Совпал с ожиданием |

Для автоматизированного тестирования [10] создан следующий модульный тест с использованием библиотеки XUnit:

* «Test\_WindDirToRmb16\_ReturnsTrue» – позитивный тест, проверяющий, что при передаче числового параметра направления ветра возвращается корректное строковое значение (код представлен листингом 4).

Листинг 4 – Код модульного теста «Test\_WindDirToRmb16\_ReturnsTrue»

|  |
| --- |
| public sealed class WindDirectorCalculatorTests  {  [Theory]  [InlineData("С", 1)]  [InlineData("С", 35)]  [InlineData("С", 36)]  [InlineData("ССВ", 2)]  [InlineData("ССВ", 3)]  [InlineData("СВ", 4)]  [InlineData("СВ", 5)]  [InlineData("ВСВ", 6)]  [InlineData("ВСВ", 7)]  [InlineData("В", 8)]  [InlineData("В", 9)]  [InlineData("В", 10)]  [InlineData("ВЮВ", 11)]  [InlineData("ВЮВ", 12)]  [InlineData("ЮВ", 13)]  [InlineData("ЮВ", 14)]  [InlineData("ЮЮВ", 15)]  [InlineData("ЮЮВ", 16)]  [InlineData("Ю", 17)]  [InlineData("Ю", 19)]  [InlineData("ЮЮЗ", 20)]  [InlineData("ЮЮЗ", 21)]  [InlineData("ЮЗ", 22)]  [InlineData("ЮЗ", 23)]  [InlineData("ЗЮЗ", 24)]  [InlineData("ЗЮЗ", 25)]  [InlineData("З", 26)]  [InlineData("З", 28)]  [InlineData("ЗСЗ", 29)]  [InlineData("ЗСЗ", 30)]  [InlineData("СЗ", 31)]  [InlineData("СЗ", 32)]  [InlineData("ССЗ", 33)]  [InlineData("ССЗ", 34)]  public void Test\_WindDirToRmb16\_ReturnsTrue(string expected, int direction)  {  //Arrange  var service = new WindDirectCalculator();  //Act  var result = service.WindDirToRmb16(direction);  //Assert  Assert.Equal(expected, result);  }  } |

Результаты модульного тестирования представлены на рисунке 4.

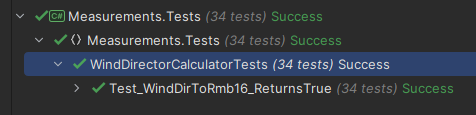


Рисунок 4 – JetBrains Rider. Вид окна с результатами прохождения модульного тестирования

Проведенное тестирование подтверждает стабильность и надежность функционирования разработанного ПО. Все заявленные функции выполняются корректно и без ошибок.

**4 Инструкция по эксплуатации программного обеспечения**

**4.1 Конфигурация программного обеспечения**

Перед началом установки ПО необходимо выполнить конфигурацию различных компонентов системы. Корневая папка проекта называется «Measurements».

Файл конфигурации клиентской части приложения расположен в «Measurements/measurements-application/.env». В нем можно настроить следующие параметры:

* «VITE\_API\_URL» – URL адрес серверной части приложения.

Файл конфигурации серверной части приложения расположен в «Measurements/Measurements.Api/appsettings.json». В нем можно настроить следующие параметры:

* «ConnectionStrings.Postgres» – строка подключения к БД,
* «Jwt.Key» – секретный ключ для подписания токенов,
* «Jwt.Issuer» – идентификатор, указывающий, кто выдал токен,
* «Jwt.Audience» – идентификатор, указывающий, для кого предназначен токен.

Файл конфигурации proxy сервера расположен в «Measurements/Nginx/nginx.conf». В нем можно настроить следующие параметры:

* «server\_name» – доменное имя, используемое для доступа к приложению,

Файл конфигурации docker compose находится в «Measurements/docker-compose.yml». В нем можно настроить следующие параметры:

* «database.enviroment.POSTGRES\_PASSWORD» – пароль для доступа к серверу БД,
* «proxy.ports» – порт, используемый для доступа к приложению.

**4.2 Установка программного обеспечения**

Процедура установки ПО на сервере значительно упрощается благодаря использованию Docker-контейнеров [11]. Применение контейнеризации позволяет существенно снизить аппаратные требования, что, в свою очередь, уменьшает необходимость в высокопроизводительном оборудовании.

Для установки приложения требуется сервер, работающий под управлением ОС Windows или Linux, с оперативной памятью объемом не менее 4 ГБ, процессором архитектуры x64 с двумя ядрами и тактовой частотой не менее 2 ГГц, а также 30 ГБ свободного дискового пространства. Также, на сервере необходимо установить Docker и плагин Docker-Compose [5]. При соблюдении всех указанных требований можно переходить к следующему этапу. Для этого необходимо перейти в корневую папку проекта c названием «Measurements», вызвать терминал и выполнить команду «docker-compose up -d». Эта команда автоматически развернет все необходимые контейнеры, обеспечивая корректную работу приложения. После успешного развертывания, приложение становится доступным по адресу, указанному в конфигурации proxy сервера, по умолчанию: «http://localhost:8080». Логин и пароль для входа по умолчанию: «Admin» и «Admin» соответственно.

**4.3 Руководство по эксплуатации**

Для обеспечения безопасности и эффективного управления информацией все страницы предоставляются исключительно авторизованным пользователям с ролями «пользователь» или «администратор». Это обеспечивает контролируемый доступ к различным функциональным возможностям приложения.

После авторизации в веб-приложении на странице «Вход» (рисунок 5), пользователь перенаправляется на страницу «Измерения», представленную на рисунке 6 и 7.

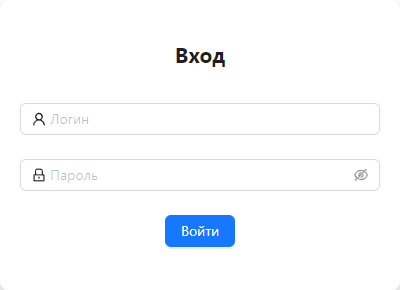


Рисунок 5 – Воздух. Вид страницы «Вход»

В левой части страницы «Измерения» находится список лабораторий. Выбрав лабораторию, пользователь увидит список городов, прикрепленных к выбранной лаборатории. После выбора города, раскрывается список станций находящихся в выбранном городе. В правой части страницы «Измерения» находятся две вкладки: «Карта» и «Данные». На вкладке «Карта» (рисунок 6) отображаются все станции, либо станции, принадлежащие выбранному городу или лаборатории, выбранным в левой части страницы. Также на вкладке «Карта» предусмотрена возможность экспорта участка карты в форматах «png» или «jpg» при помощи кнопки «Экспортировать». На вкладке «Данные» (рисунок 7) отображаются в табличном виде данные всех измерений. После выбора объекта в списке слева, отображаются измерения, относящиеся только к выбранному объекту. Также на вкладке «Данные» доступна возможность экспорта измерений в файлы форматов «xlsx» и «csv» при помощи кнопки «Экспортировать», а также возможность фильтрации данных за определенный период времени.

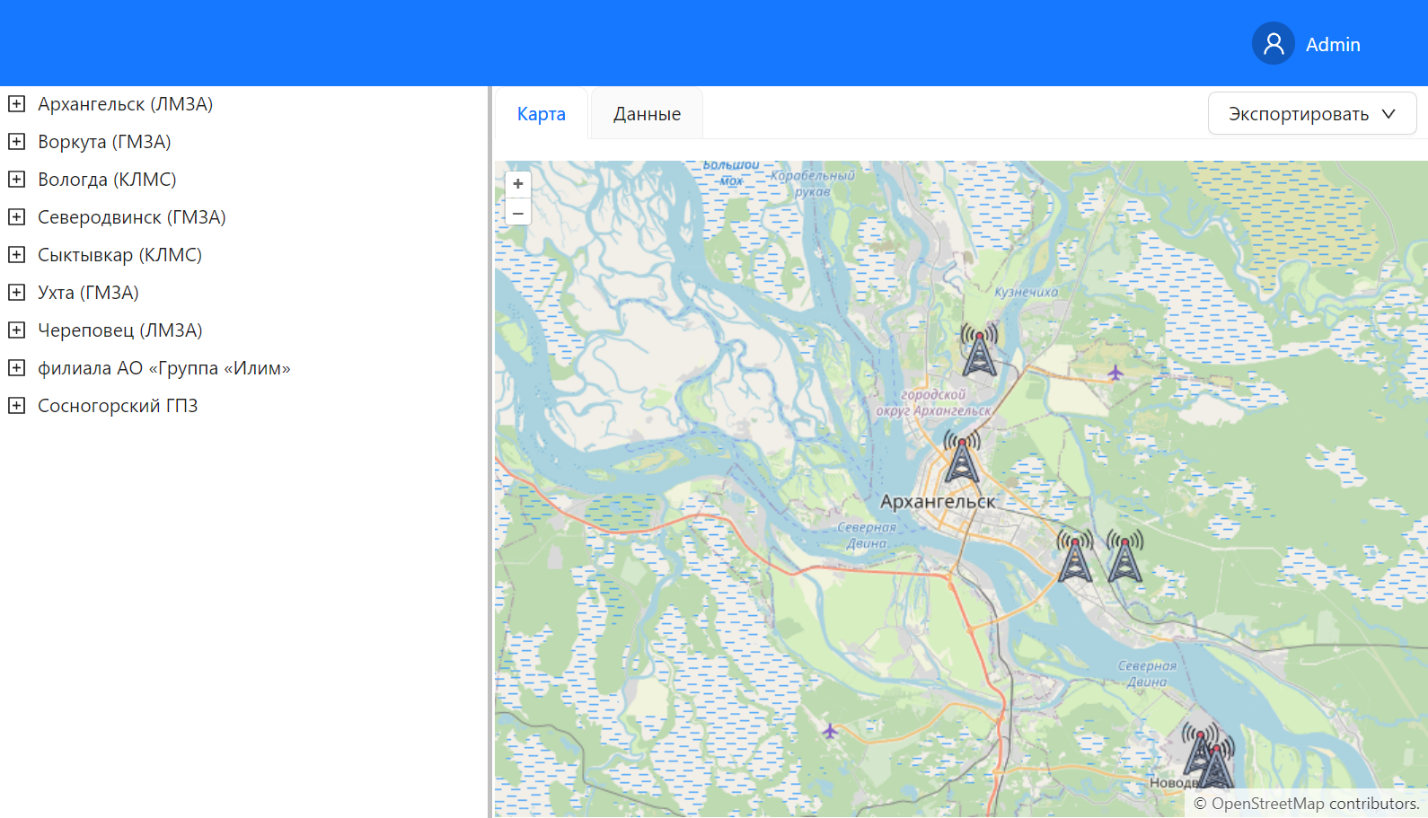


Рисунок 6 – Воздух. Вид страницы «Измерения» и вкладки «Карта»

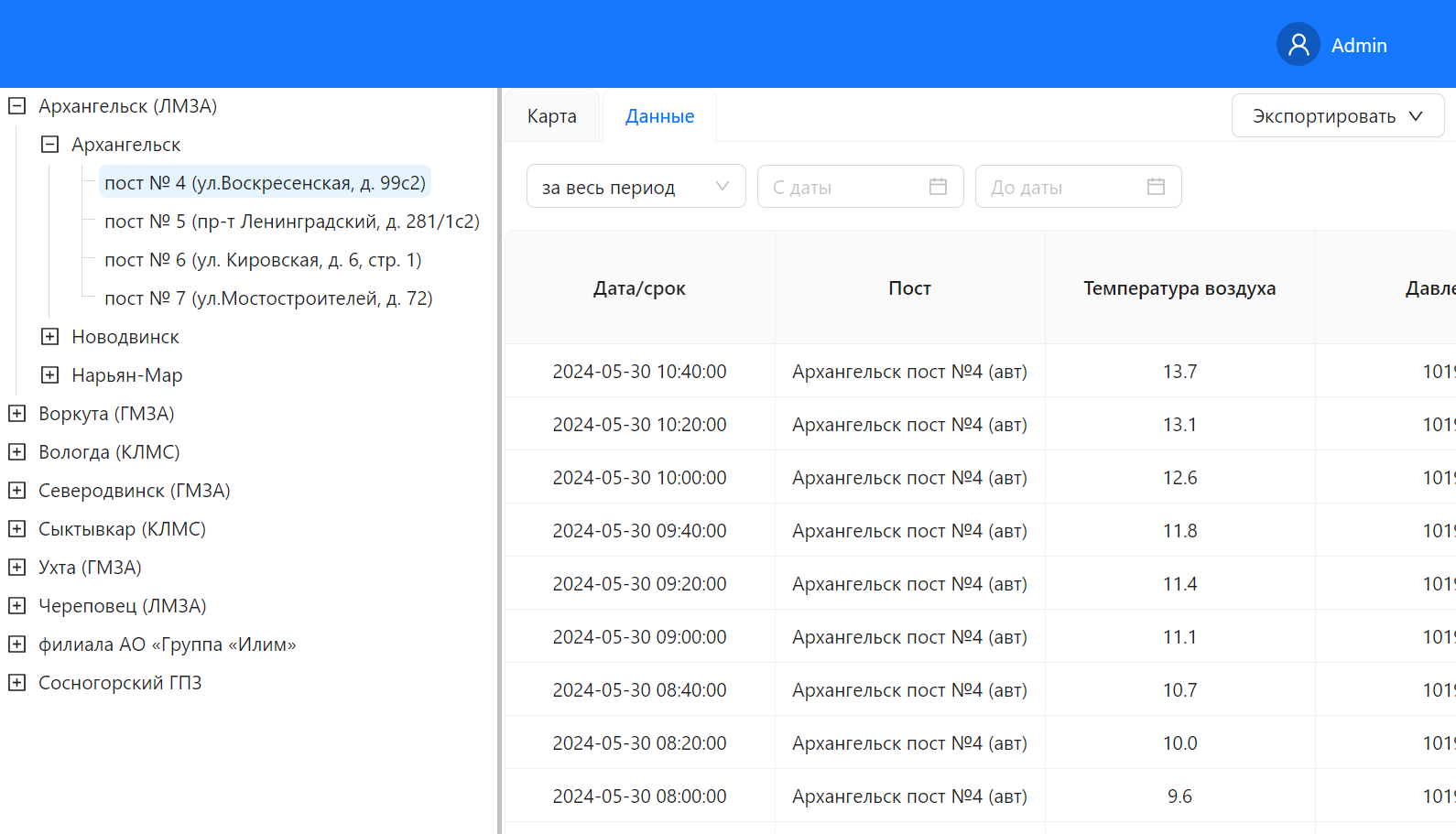


Рисунок 7 – Воздух. Вид страницы «Измерения» и вкладки «Данные»

Страница «Пользователи» (рисунок 8) доступна исключительно пользователям с ролью «Администратор». Страница содержит таблицу пользователей, включая функциональность поиска пользователя по логину, фильтрации пользователей по ролям, удаление информации о пользователе при помощи кнопки с иконкой корзины и возможность блокировки и разблокировки пользователя с помощью переключателя в колонке «Заблокирован».

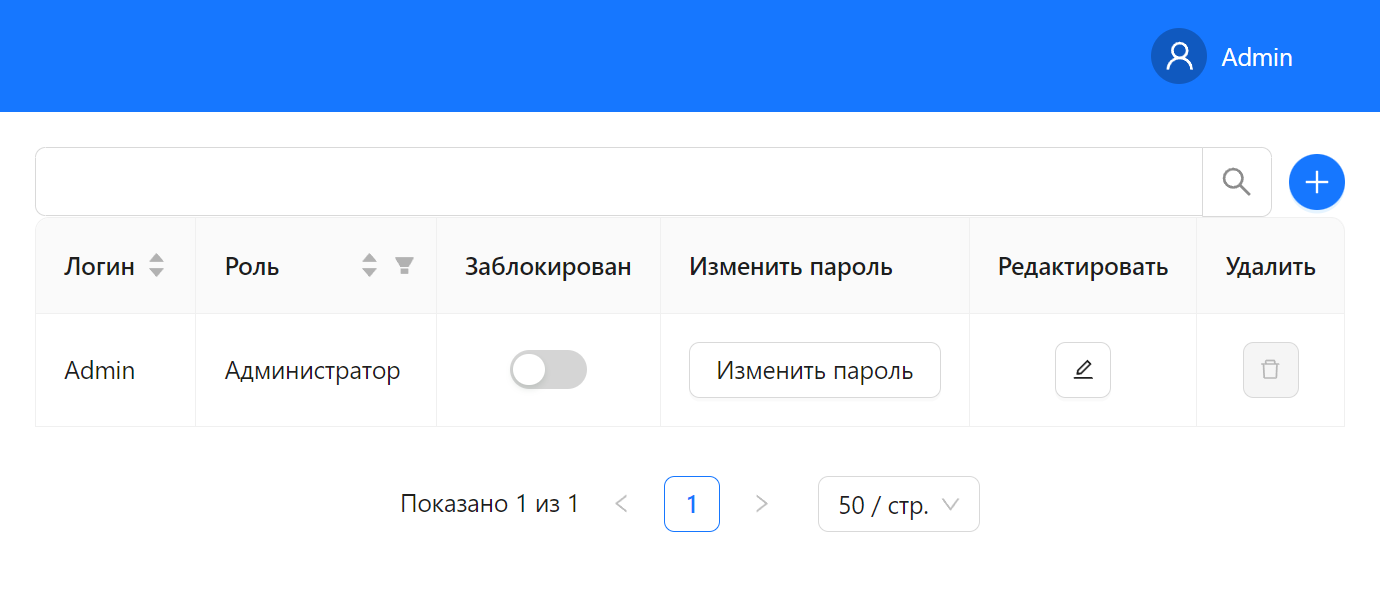


Рисунок 8 – Воздух. Вид страницы «Пользователи»

Для изменения пароля пользователя необходимо нажать кнопку «Изменить пароль», при нажатии открывается модальное окно «Смена пароля», представленное рисунком 9.

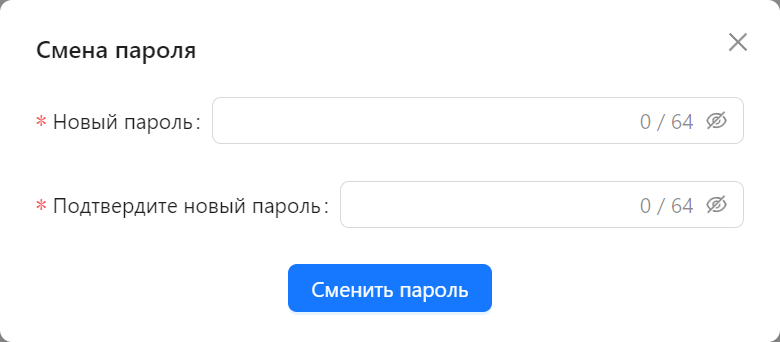


Рисунок 9 – Воздух. Модальное окно «Смена пароля»

Для добавления информации о новом пользователе необходимо нажать кнопку с иконкой плюса, находящуюся справа от строки поиска, при нажатии открывается модальное окно «Добавление пользователя», представленное рисунком 10. Для изменения данных пользователя, включая доступные для просмотра станции и измерения, необходимо нажать кнопку с иконкой карандаша, при нажатии открывается модальное окно «Обновление пользователя», представленное рисунком 11.



Рисунок 10 – Воздух. Модальное окно «Добавление пользователя (Данные)»

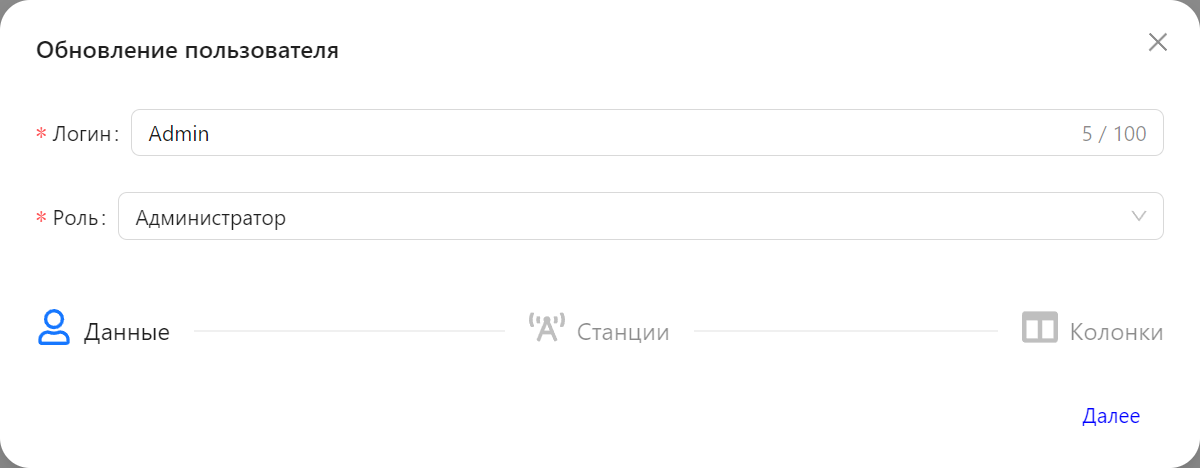


Рисунок 11 – Воздух. Модальное окно «Обновление пользователя (Данные)»

Модальные окна «Добавление пользователя» и «Обновление пользователя» включают три этапа. На первом этапе требуется ввести логин и выбрать роль пользователя, а при добавлении информации о новом пользователе, а также указать и подтвердить пароль. После этого необходимо нажать кнопку «Далее» для перехода ко второму этапу, для выбора доступных пользователю станций (рисунок 12). Для удобства выбора станций предусмотрена функциональность поиска по названию, лаборатории, городу, адресу, а также возможность фильтрации станций по лаборатории и городу.

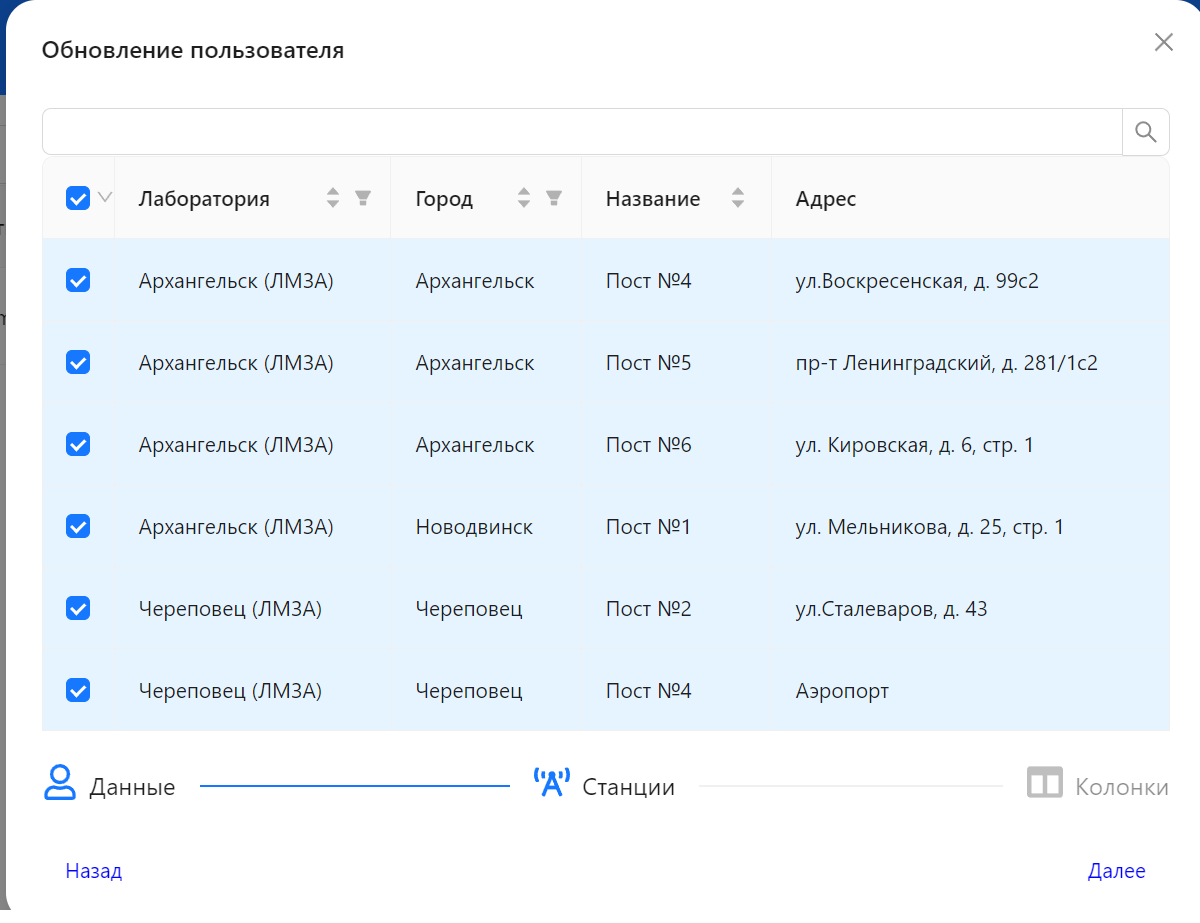


Рисунок 12 – Воздух. Модальное окно «Обновление пользователя (Станции)»

После выбора станций необходимо нажать кнопку «Далее» для перехода к последнему этапу, где необходимо указать доступные для просмотра измерения (рисунок 13). Для удобства выбора измерений предусмотрена возможность поиска по названию. После заполнения доступных измерений необходимо нажать кнопку «Готово». Для возвращения на предыдущий этап предусмотрена кнопка «Назад».

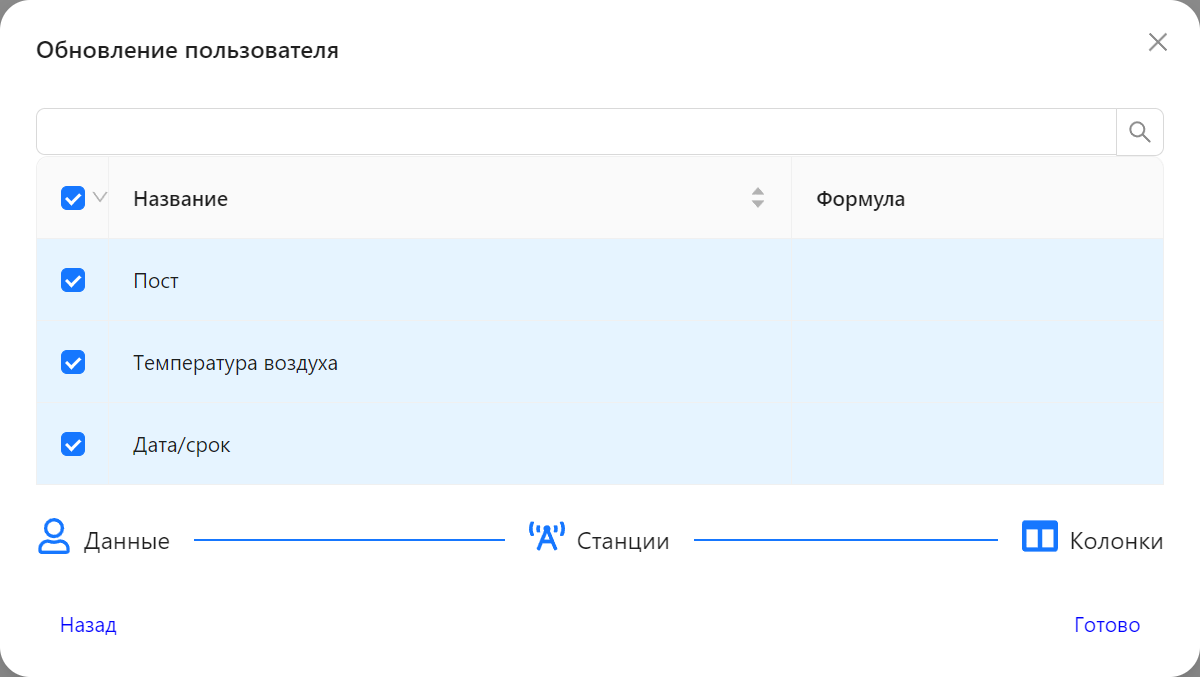


Рисунок 13 – Воздух. Модальное окно «Обновление пользователя (Колонки)»

Страница «Станции» (рисунок 14) доступна исключительно пользователям с ролью «Администратор». Страница содержит таблицу станций, включая функциональность отключения сбора данных со станции, включения сбора данных со станции с помощью переключателя в колонке «Опрашивается». Для удобства пользователей реализована функциональность поиска по названию, лаборатории, городу и адресу, а также фильтрация станций по лаборатории и городу. Справа от строки поиска расположена кнопка «Синхронизировать», при нажатии на которую происходит поиск новых станций в программный продукт «АСОИЗА+» и обновление неактуальных данных станций. Найденные станции добавляются, а устаревшие данные обновляются в приложении «Воздух». Результат выполнения синхронизации данных станций с программный продукт «АСОИЗА+» отображен на рисунке 15.

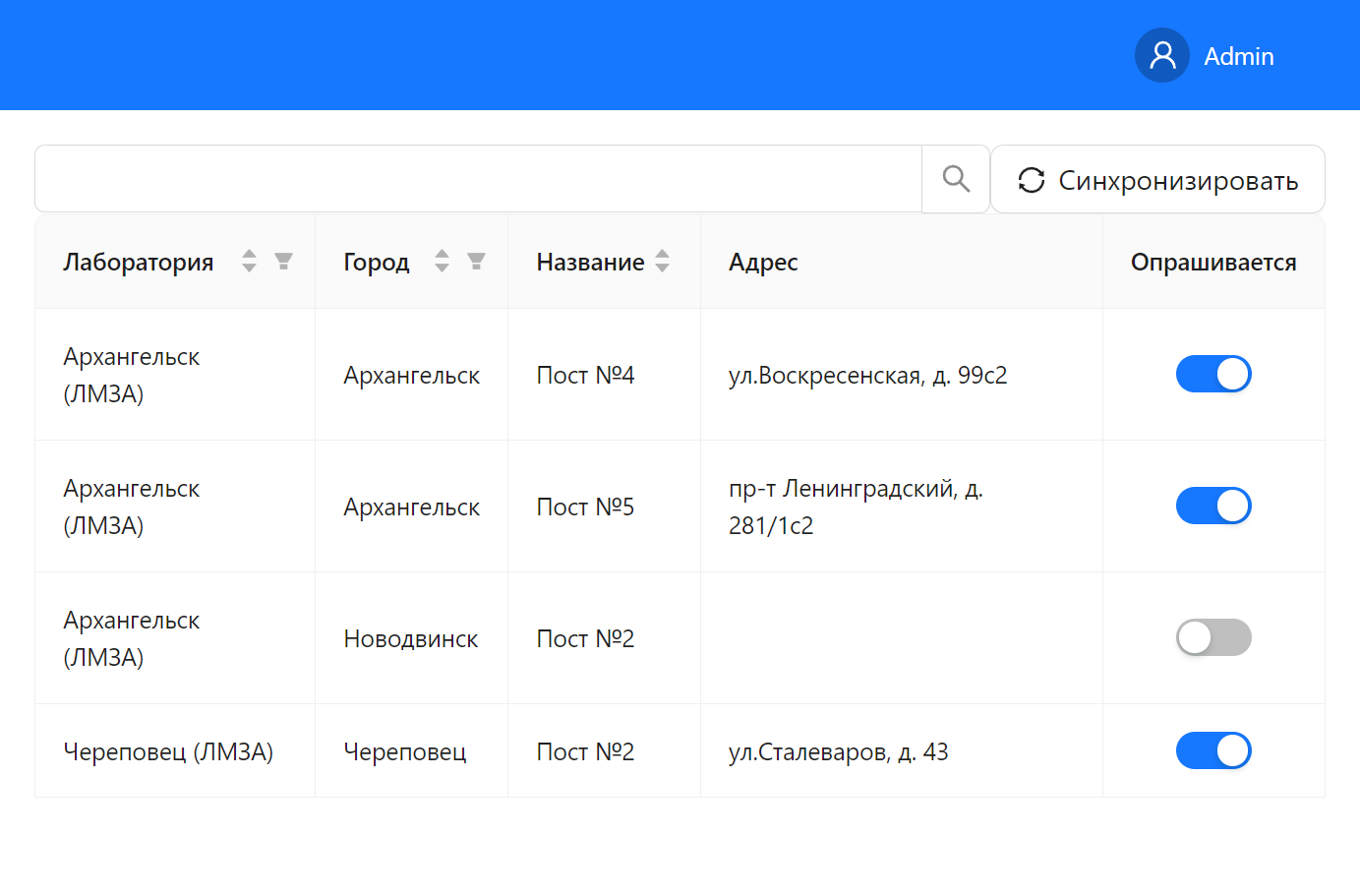


Рисунок 14 – Воздух. Вид страницы «Станции»

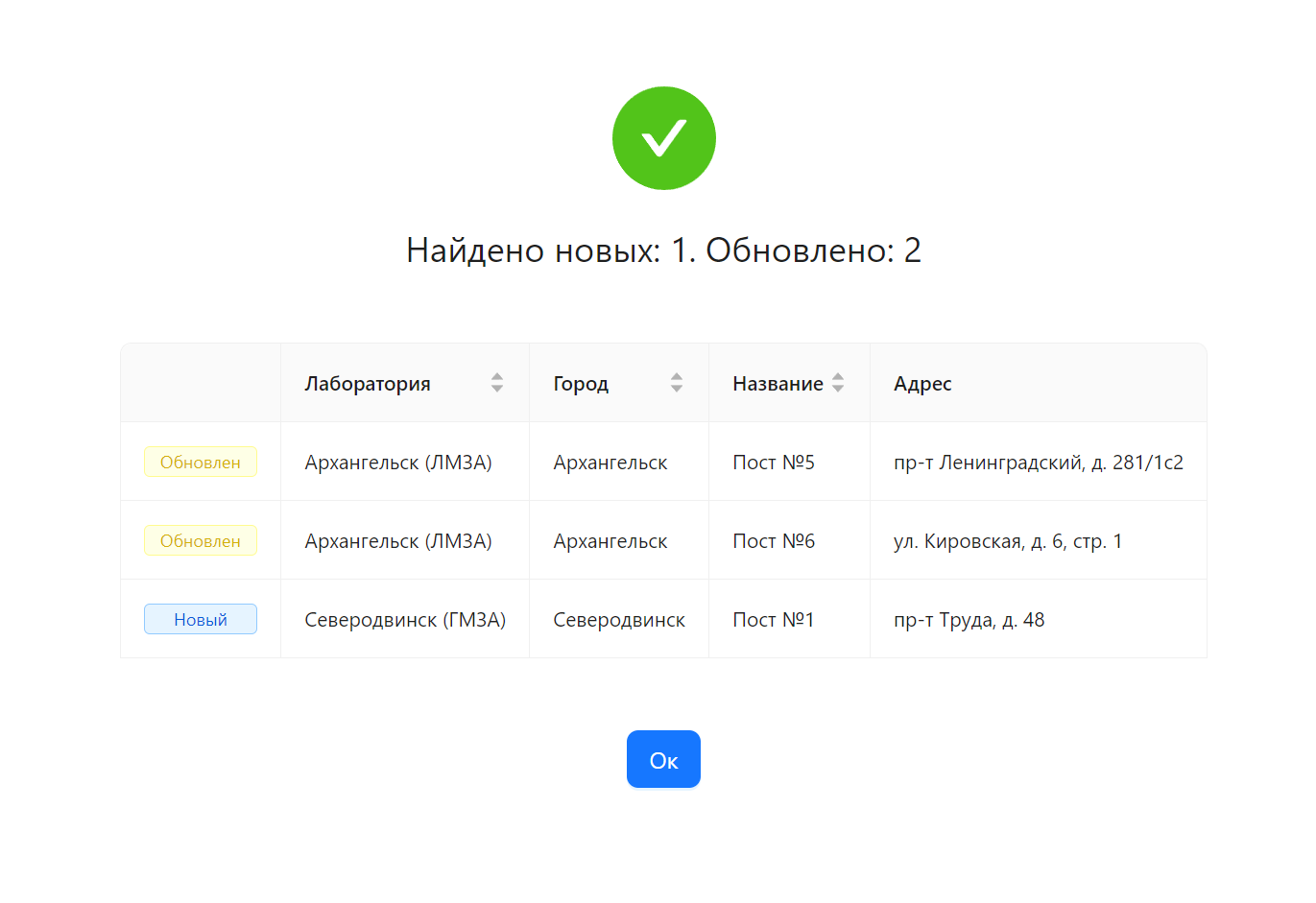


Рисунок 15 – Воздух. Вид страницы «Результат синхронизации станций»

Страница «Конфигурация» доступна только пользователям с ролью «Администратор». Она включает три секции: «Данные для входа в АСОИЗА+» (рисунок 16), «Сбор данных» (рисунок 17) и «Хранение данных» (рисунок 18), а также кнопку «Сохранить». В разделе «Данные для входа в АСОИЗА+» необходимо указать логин и пароль для доступа к программному продукту «АСОИЗА+». В секции «Сбор данных» задается интервал сбора данных со станций (в минутах) и период, через который автоматически отключается сбор данных для оптимизации ресурсов и ускорения обновления данных в системе.

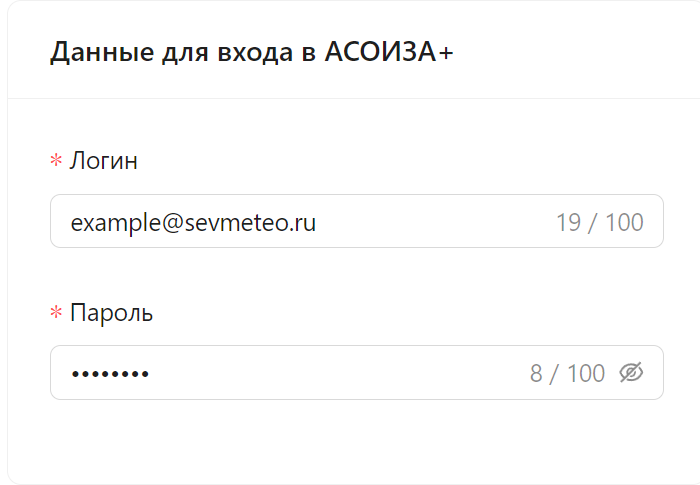


Рисунок 16 – Воздух. Вид блока «Данные для входа в АСОИЗА+»

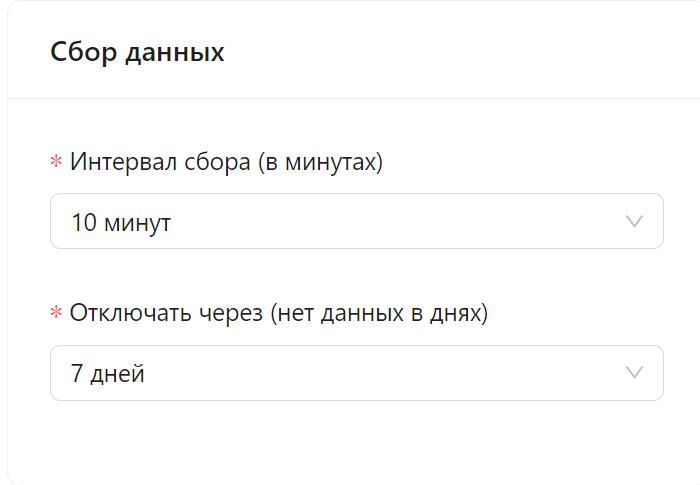


Рисунок 17 – Воздух. Вид блока «Сбор данных»

В разделе «Хранение данных» указывается срок хранения собранных данных и интервал чистки устаревших данных (в днях).

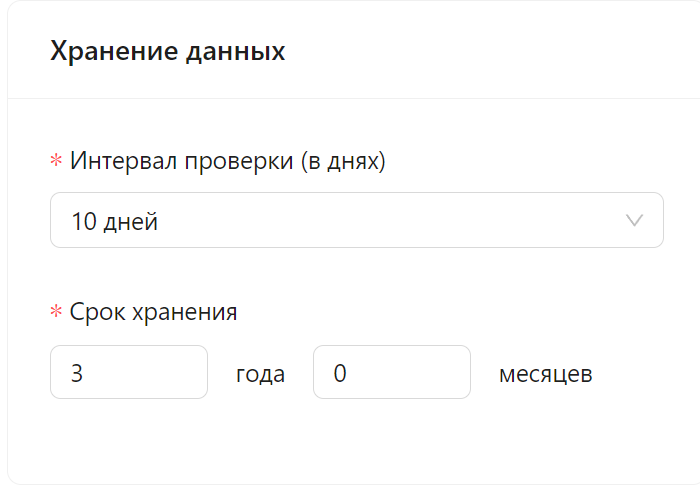


Рисунок 18 – Воздух. Вид блока «Хранение данных»

Страница «Статистика» (рисунок 20) доступна только пользователям с ролью «Администратор». На этой странице представлена информация о количестве станций, с которых собираются данные, дате и времени последнего сбора данных, времени до следующего сбора, интервале опроса станций, а также график, отображающий информацию о каждом сборе данных. График включает дату и время опроса, количество опрошенных станций, количество собранных измерений и длительность опроса. Пример отображения представлен рисунком 19.

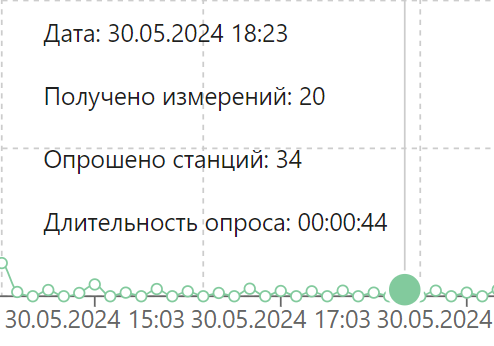


Рисунок 19 – Воздух. Вид подсказки у точки графика

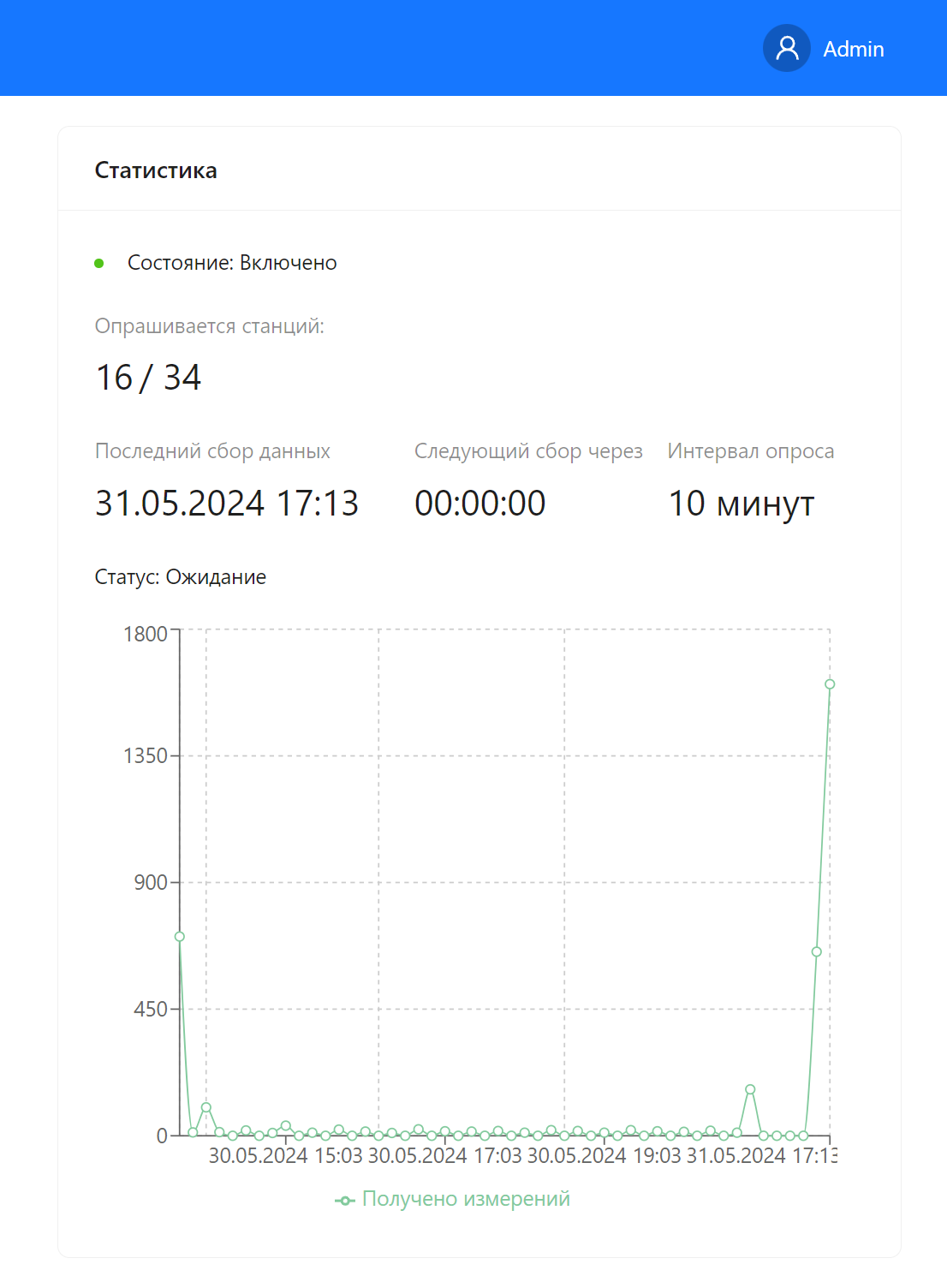


Рисунок 20 – Воздух. Вид страницы «Статистика»

**5 Определение затрат на разработку**

**5.1 Определение затрат на разработку**

Определение затрат на разработку является ключевым элементом дипломного проектирования. Оно позволяет провести комплексную оценку целесообразности и эффективности планируемой разработки ПО.

Для разработки системы автоматического сбора данных необходимо учесть ряд важных факторов, включая стоимостные оценки и потенциальные риски. Затраты на разработку ПО формируются из расходов по оплате машинного времени и общих затрат.

Затраты на создание программного продукта Зспп, руб. определяются по формуле

, (1)

где – затраты на создание программного продукта;

– затраты на оплату машинного времени;

– общие затраты.

Трудоемкость разработки системы автоматизации 𝑡, чел. ч, определяется по формуле

, (2)

где – затраты труда на подготовку описания задачи;

– затраты труда на исследование алгоритма решения задачи;

– затраты труда на разработку блок-схемы алгоритма;

– затраты труда на программирование по готовой блок-схеме;

– затраты труда на отладку программы ЭВМ;

– затраты труда на подготовку документации.

Коэффициент, учитывающий новизну и сложность программы (с) определяется исходя из таблицы 2 на пересечении групп сложности и степени новизны. Программный продукт по степени новизны может быть отнесен к одной из четырех групп:

* группа А – разработка принципиально новых задач,
* группа Б – разработка оригинальных программ,
* группа В – разработка программ с использованием типовых решений,
* группа Г – разовая типовая задача.

По степени сложности программные продукты могут быть отнесены к одной из трех групп:

* 1 – алгоритм оптимизации и моделирования систем,
* 2 – задачи учета, отчетности и статистики,
* 3 – стандартные алгоритмы.

Таблица 2 – Значение коэффициентов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Язык программирования | Группа сложности | Степень новизны | | | | Коэффициент, B |
| А | Б | В | Г |
| Высокого уровня | 1 | 1,38 | 1,26 | 1,15 | 1,20 | 1,20 |
| 2 | 1,30 | 1,19 | 1,08 | 0,65 | 1,35 |
| 3 | 1,20 | 1,10 | 1,00 | 0,60 | 1,50 |
| Низкого уровня | 1 | 1,58 | 1,45 | 1,32 | 0,79 | 1,20 |
| 2 | 1,49 | 1,37 | 1,24 | 0,74 | 1,35 |
| 3 | 1,38 | 1,26 | 1,15 | 0,69 | 1,50 |

Коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнений квалификации разработчика (К) можно по данным таблицы 3.

Таблица 3 – Коэффициент квалификации разработчика

|  |  |
| --- | --- |
| Опыт работы | Коэффициент квалификации |
| До двух лет | 0,80 |
| 2-3 года | 1,00 |
| 3-5 лет | 1,10-1,20 |
| 5-7 лет | 1,30-1,40 |
| Более 7 лет | 1,50-1,60 |

Составление затрат вычисляется при помощи условного числа операторов. Условное число операторов Q, ед, в программе определяется по формуле

(3)

где q – число исходных команд;

с – коэффициент, учитывающий новизну и сложность программы;

p – коэффициент коррекции программы в ходе ее разработки.

В разработанной системе автоматического сбора данных число операторов составляет около 586.

Коэффициент, учитывающий новизну и сложность программы определяется исходя из таблицы 2. Разработанный программный продукт по степени новизны относится к разработке программ с использованием типовых решений, а по степени сложности алгоритма стандартные алгоритмы. Тогда, по таблице 2 коэффициент c = 1,15 и коэффициент   
B = 1,20.

С учётом того, что задача поставлена достаточно чётко, коэффициент p принимается равным 0,05. Условное число операторов, согласно формуле (3), составляет

Затраты труда на подготовку описания задачи to, чел. ч, точно определить невозможно, т.к. это связано с творческим характером работы. Данное значение принимается равным 50 чел. ч

Затраты труда на изучение описания задачи с учётом уточнения описания и квалификации программиста , чел. ч, определяются по формуле

, (4)

где Q – условное число операторов;

K – коэффициент квалификации разработчиков алгоритма и программ;

B – коэффициент увеличения затрат труда вследствие недостаточного или некачественного описания задачи.

По таблице 3 для работающих до двух лет K = 0,80.

Необходимо вычислить затраты труда на различных стадиях разработки программного продукта.

Затраты труда на разработку алгоритма решения задачи , чел. ч, определяются по формуле

, (5)

Затраты труда на разработку диаграмм алгоритма решения задачи , чел. ч, определяются по формуле

, (6)

Затраты труда на составление программы по готовой блок-схеме , чел. ч, определяются по формуле

, (7)

Затраты труда на отладку программы на ЭВМ при комплексной отладке , чел. ч, определяются по формуле

, (8)

где затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи.

Затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи , чел. ч, определяются по формуле

, (9)

Расчет затрат труда на отладку программы на ЭВМ при комплексной отладке выполняется по формуле (8)

Затраты труда на подготовку документации по задаче , чел. ч, определяются по формуле

, (10)

где – затраты труда на подготовку материалов рукописи;

– затраты на редактирование, печать и оформление документации.

Затраты труда на подготовку материалов рукописи , чел. ч, определяются по формуле

, (11)

Затраты на редактирование, печать и оформление документации , чел. ч, определяются по формуле

, (12)

Расчёт затрат труда на подготовку документации по задаче выполняется по формуле (10)

Трудоёмкость разработки программного продукта, согласно формуле (2), составляет

Трудоемкость разработки составила 122,60 чел. ч, что составляет 16 рабочих дней при восьмичасовом рабочем дне. Структура трудозатрат предоставлена на рисунке 21.

Рисунок 21 – Диаграмма общей трудоемкости разработанного ПО

Этап подготовки описания задачи является критически важным в процессе разработки ПО, так как он оказывает значительное влияние на общую трудоемкость проекта. Тщательное и детализированное описание задачи позволяет избежать недоразумений и ошибок на последующих этапах разработки, что, в свою очередь, способствует более эффективному использованию ресурсов и времени, а также отсутствию необходимости внесения значительных изменений в уже реализованные компоненты системы. Таким образом, качественная подготовка описания задачи является залогом успешной и экономически эффективной реализации программного продукта.

Далее необходимо вычислить все финансовые затраты на разработку проекта.

Затраты на оплату машинного времени при отладке программы , руб. определяются по формуле

, (13)

где – цена машино-часа арендного времени, руб/час;

– фактическое время отладки программы на ЭВМ, чел. ч;

Фактическое время отладки , чел. ч, определяется по формуле

, (14)

Цена машино-часа , руб/ч, определяется по формуле

, (15)

где – затраты на эксплуатацию ЭВМ, руб;

– действительный месячный фонд времени ЭВМ, ч.

Действительный месячный фонд времени ЭВМ , ч, определяется по формуле

, (16)

Общее количество календарных дней в месяце , число праздничных и выходных дней в месяце в соответствии со статьей 112 трудового кодекса РФ [4].

Время простоя в профилактических работах определяется как еженедельная профилактика по 4 часа.

Выполняется расчет действительного месячного фонда времени ЭВМ по формуле (16)

Затраты на эксплуатацию ЭВМ , руб. определяются по формуле

, (17)

где – издержки на амортизацию, руб.;

– издержки на электроэнергию, потребляемую ЭВМ, руб.

Согласно ч. 2 ст.256 и ст. 257 п. 1 налогового кодекса РФ амортизируемым имуществом признается имущество со сроком полезного использования более 12 месяцев и первоначальной стоимостью более 100,00 тыс. руб. [1].

Так как ПК, на котором велась работа по разработке системы автоматизации приобретен за 71 344,56 руб. то стоимость годовых амортизационных отчислений не считается, Зам принимается за 0.

Стоимость электроэнергии, потребляемой за месяц, , руб. определяется по формуле

, (18)

где – суммарная мощность ЭВМ, кВт;

– стоимость 1 кВт/ч электроэнергии, руб.

Согласно техническому паспорту ЭВМ, потребление электроэнергии составляет 0,426 кВт.

Исходя из тарифа 6,06 кВт/ч [13], который используется, стоимость электроэнергии, потребляемой за месяц по формуле (18), составляет

Затраты на эксплуатацию ЭВМ, согласно формуле (17), составляют

Цена машино-часа, согласно формуле (14), составляет

Затраты на оплату машинного времени при отладке программы, согласно формуле (13), составляют

Общие затраты , руб. определяются по формуле

, (19)

где – издержки на заработную плату, руб.;

– издержки на страховые взносы с оплаты труда, руб.;

– издержки на прочие и накладные расходы, руб.

Заработная плата работника , руб. определяется по формуле

, (20)

где  – коэффициент, учитывающий районную надбавку на территориях, приравненных к Крайнему Северу,

– коэффициент, учитывающий северную надбавку для работающих в местах, приравненных к Крайнему Северу.

Районный коэффициент в г. Архангельске составляет 20% от основной заработной платы, а выплаты за выслугу лет, проработанных на территории, приравненной к территории Крайнего Севера – 50% от основной заработной платы. За отработанный месяц заработная плата должна быть не меньше минимального размера оплаты труда, согласно ст.1 Федерального закона от 19.06.2000 №82-ФЗ «О минимальном размере оплаты труда» в редакции Федерального закона от 27.11.2023 № 548-ФЗ [3], без учета районных коэффициентов и северных надбавок в г. Архангельске составляет 19 242,00 руб. Оклад программистов, работающих в г. Архангельске, принимается за 19 242,00 руб.

Так как разработка проекта ведется на территории с особыми климатическими условиями, то на основании статьи 317 трудового кодекса РФ [4] и ст. 11 закона РФ от 19.02.1993 г. № 4520-1 [2] к окладу применяются районные коэффициенты и надбавки.

Расчёт заработной платы работника выполняется по формуле (20)

Страховые взносы с платы труда , руб. определяются по формуле

, (21)

где – фонд оплаты труда, руб.;

– актуальный размер страховых взносов, %.

Издержки на страховые взносы с оплаты труда определяются гл. 34 ст. 425 налогового кодекса РФ [1]. Так как число работников соответствует одному, то будет равным , а размер отчислений составляют 30% от суммы заработной платы, без учета взносов на травматизм, так как эта сумма зависит от опасности производства.

Стоимость отчислений страховых взносов с оплаты труда, согласно формуле (21), составляет

Прочие затраты , руб. принимаются в размере 10% в общей сумме затрат и определяются по формуле

, (22)

Общие затраты, согласно формуле (19), составляют

Затраты на создание программного продукта, согласно формуле (1), составляют

В результате произведённых расчётов затраты на создание программного продукта составляют 47 666,99 руб.

На рисунке 22 представлена диаграмма затрат.

Рисунок 22 – Диаграмма затрат

Трудоемкость разработки системы автоматического сбора данных «Воздух» составила 122,60 чел. ч, при этом затраты на реализацию проекта составили 47 666,99 рублей. Для сравнения, средняя стоимость аналогичных систем на рынке варьируется от 80 000 до 150 000 рублей. Таким образом, можно сделать вывод, что разработка данной системы выполнена более эффективно, с использованием меньшего количества ресурсов, что свидетельствует о высокой экономической эффективности проекта.

**5.2 Анализ экономического обоснования и рисков проекта**

Для эффективной разработки системы автоматического сбора данных «Воздух» необходимо сделать анализ рисков проекта и разработать стратегию управления ими. Анализ рисков позволит определить потенциальные проблемы, которые возможно возникнут при разработке проекта, а также оценить вероятность их появления и уровень воздействия на проект.

В процессе анализа учтены затраты на разработку и внедрение системы. Результаты показали, что внедрение системы автоматического сбора данных «Воздух» принесет значительные преимущества. В частности, автоматизация процессов мониторинга показателей качества воздуха позволит более удобно и менее затратно анализировать наличие загрязнений в атмосфере и своевременно устранять их.

Проект является некоммерческим, предназначен для мониторинга данных о загрязнении воздуха, включая показатели температуры, давления и общие показатели качества воздуха. Предложенное решение полностью удовлетворяет интересам заказчика.

**6 Охрана труда и техника безопасности при работе на ПК**

**6.1 Общие требования охраны труда**

К самостоятельной работе на ПК допускаются лица, прошедшие предварительные и периодические медицинские осмотры, вводный инструктаж по охране труда, обучение безопасным приемам работы на ПК и первичный инструктаж на конкретном рабочем месте.

При работе на ПК работнику необходимо учитывать возможное воздействие следующих вредных и опасных производственных факторов: повышенная температура поверхностей ПК, повышенное напряжение в электрической цепи, уровень статического электричества, уровень электромагнитных излучений, напряженность электрического поля, недостаточная освещенность рабочей зоны, повышенная яркость света, повышенная контрастность, зрительное напряжение, длительные статические нагрузки и монотонность трудового процесса.

При работе на ПК работник обязан соблюдать производственную и технологическую дисциплину, регулировать режим труда и отдыха, выполнять задачи по должностной инструкции, поддерживать порядок на рабочем месте, сообщать о неисправностях ПК и электропитания руководителю и соблюдать требования пожарной и электробезопасности.

За нарушение требований охраны труда работник привлекается к ответственности в порядке, установленном законодательством РФ.

**6.2 Требования охраны труда перед началом работы**

Перед началом работы на ПК необходимо проветрить помещение, настроить освещение на рабочем месте, убедиться в правильности расположения ПК для предотвращения неудобных поз и напряжений тела, а также убрать все посторонние предметы с рабочего места и освободить доступ к нему.

Необходимо проверить исправность всех кабельных соединений, убедиться в отсутствии пыли на экране монитора и клавиатуре, при необходимости протереть их специальной салфеткой.

**6.3 Требования охраны труда во время работы**

При подключении ПК к сети электропитания необходимо использовать только штатные сетевые кабели, следить за тем, чтобы кожухи были закрыты, а заземление присутствовало.

При включении ПК следует придерживаться следующего порядка действий: сначала включить сетевой фильтр, затем источник бесперебойного питания, системный блок и монитор. После включения ПК и запуска ПО, необходимо убедиться в отсутствии дрожания и мерцания изображения на экране монитора, а также настроить яркость, контрастность, цвет, размер символов, и фон экрана для наиболее комфортного и четкого восприятия изображения.

Во время работы на ПК необходимо избегать натягивания, скручивания, перегиба и пережима шнуров электропитания, предотвращать попадание предметов на шнуры, избегать попадания влаги на поверхность ПК, не прикасаться к задней панели системного блока при включенном питании, не оставлять включенный ПК без присмотра, не загромождать рабочее место, не проводить самостоятельный ремонт ПК, не закрывать вентиляционные отверстия, не часто отключать и включать ПК, соблюдать расстояние от глаз до экрана (50-70 см), не превышать двух часов непрерывной работы без перерыва.

**6.4 Требования охраны труда по окончании работы**

Отключить ПК от электросети сухими руками, после чего, привести рабочее место в порядок.

**6.5 Требования охраны труда в аварийных ситуациях**

В случае возникновения аварийной ситуации на рабочем месте необходимо немедленно прекратить выполнение задач и уведомить об этом своего непосредственного руководителя или его заместителя. Под руководством начальства работник должен участвовать в устранении аварии, если это не угрожает его здоровью и жизни. При необходимости следует покинуть опасную зону.

При обнаружении неисправностей в работе ПК или электросети необходимо немедленно прекратить работу, отключить ПК от сети и сообщить об этом своему непосредственному руководителю или его заместителю. Возобновление работы возможно только после полного устранения неисправностей.

В случае возгорания необходимо немедленно прекратить работу, отключить ПК и уведомить об этом руководителя. Затем следует попытаться потушить огонь с помощью углекислотного или порошкового огнетушителя. Если потушить огонь не удается, необходимо покинуть опасную зону через эвакуационные выходы согласно плану эвакуации.

При резком ухудшении состояния здоровья или получении травмы необходимо прекратить работу и сообщить об этом своему непосредственному руководителю или его заместителю. При необходимости следует воспользоваться аптечкой.

**ЗАКЛЮЧЕНИЕ**

В результате дипломного проектирования разработана система автоматического сбора данных «Воздух», предназначенная для автоматического сбора данных в режиме реального времени с программного продукта «АСОИЗА+», а также визуализации этих данных для дальнейшего анализа специалистами.

Разработанное ПО введено в эксплуатацию в ФГБУ «Северное УГМС» и активно используется аэрохимиками в лаборатории загрязнения атмосферного воздуха.

В ходе дипломного проектирования решены следующие задачи:

* собраны и проанализированы функциональные и эксплуатационные требования к ПО,
* описана работа программного модуля с помощью UML,
* выбран состав программных и технических средств для разрабатываемого ПО,
* выделены объекты и атрибуты БД в соответствии с заданием,
* спроектирована БД,
* разработана БД,
* созданы запросы к БД,
* спроектирована архитектура ПО,
* спроектирован интерфейс пользователя,
* реализован интерфейс пользователя,
* разработаны программные модули системы автоматического сбора данных «Воздух»,
* выполнено тестирование и отладка разработанного ПО,
* разработана справочная документация,
* описан процесс установки разработанного ПО,
* проведена организация защиты ПО компьютерных систем программными средствами.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Российская Федерация. Законы. Налоговый Кодекс Российской Федерации. Часть вторая : Федеральный закон № 117-ФЗ : [принят Государственной думой 19 июля 2000 года : одобрен Советом Федерации 26 июля 2000 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: [https://www.consultant.ru/document/cons\_doc\_LAW\_28165/](https://www.consultant.ru/document/cons_doc_LAW_28165/%20) (дата обращения: 22.04.2024).
2. Российская Федерация. Законы. О государственных гарантиях и компенсациях для лиц, работающих и проживающих в районах Крайнего Севера и приравненных к ним местностях : Федеральный закон № 4520-1 : [принят и одобрен Президентом Российской Федерации Б. Ельциным 19 февраля 1993 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: [https://www.consultant.ru/document/cons\_doc\_LAW\_1786/](https://www.consultant.ru/document/cons_doc_LAW_1786/%20) (дата обращения: 03.05.2024).
3. Российская Федерация. Законы. О минимальном размере оплаты труда : Федеральный закон № 82-ФЗ : [принят Государственной Думой 2 июня 2000 года : одобрен Советом Федерации 7 июня 2000 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: <https://www.consultant.ru/document/cons_doc_LAW_27572/> (дата обращения: 26.04.2024).
4. Российская Федерация. Законы. Трудовой кодекс Российской Федерации : Федеральный закон № 197-ФЗ : [принят Государственной Думой 21 декабря 2001 года : одобрен Советом Федерации 26 декабря 2001 года]. – Текст : электронный // КонсультантПлюс : [сайт]. – 2024. – URL: <https://www.consultant.ru/document/cons_doc_LAW_34683/> (дата обращения: 02.05.2024).
5. Гадзурас, Э. Docker Compose для разработчика / Э. Гадзурас. – Москва : ДМК Пресс, 2023. – 221 с. – URL: <https://znanium.ru/catalog/product/2150533> (дата обращения: 15.05.2024). – Режим доступа: по подписке. – Текст : электронный.
6. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. – 2-е изд., перераб. и доп. – Москва : ФОРУМ : ИНФРА–М, 2021. – 416 с. – (Cреднее профессиональное образование). – URL: <https://znanium.ru/catalog/product/1190668> (дата обращения: 23.04.2024). – Режим доступа: по подписке. – Текст : электронный.
7. Йонге де, Д. NGINX. Книга рецептов : практическое руководство / Д. де Йонге ; пер. с анг. Д. А. Беликова. – Москва : ДМК Пресс, 2020. – 176 с. – URL: [https://znanium.com/catalog/product/1094952](https://znanium.com/catalog/product/1094952%20) (дата обращения: 14.05.2024). – Режим доступа: по подписке. – Текст : электронный.
8. Карпова, Т. С. Базы данных: модели, разработка, реализация : учебное пособие / Т. С. Карпова. – Москва : ИНТУИТ, 2016. – 288 с. – URL: <https://znanium.ru/catalog/product/2138291> (дата обращения: 25.04.2024). – Режим доступа: по подписке. – Текст : электронный.
9. Лок, Э. ASP.Net Core в действии : практическое руководство / Э. Лок ; пер. с анг. Д. А. Беликова. – Москва : ДМК Пресс, 2021. – 906 с. – URL: <https://znanium.ru/catalog/product/2155886> (дата обращения: 03.05.2024). – Режим доступа: по подписке. – Текст : электронный.
10. Морозова, Ю. В. Тестирование программного обеспечения : учебное пособие / Ю. В. Морозова. – Томск : Эль-Контент, 2019. – 120 с. – URL: <https://znanium.com/catalog/product/1845910> (дата обращения: 23.05.2024). – Режим доступа: по подписке. – Текст : электронный.
11. Моуэт, Э. Использование Docker / Э. Моуэт ; пер. с англ. А.В. Снастина ; под науч. ред. А. А. Маркелова. – Москва : ДМК Пресс, 2017. – 354 с. – URL: <https://znanium.com/catalog/product/1027859> (дата обращения: 24.05.2024). – Режим доступа: по подписке. – Текст : электронный.
12. Назаров, С. В. Архитектура и проектирование программных систем : монография / С. В. Назаров. – 2-е изд., перераб. и доп. – Москва : ИНФРА-М, 2023. – 374 с. – (Научная мысль). – URL: <https://znanium.com/catalog/product/1895672> (дата обращения: 20.05.2024). – Режим доступа: по подписке. – Текст : электронный.
13. Официальное опубликование правовых актов : региональный цифровой ресурс : Постановление № 81 – э / 5 Об установлении цен (тарифов) на электрическую энергию : сайт / Правительства Архангельской Области. – Архангельск, 2023 – 2024. – URL: <http://publication.pravo.gov.ru/document/2901202312250052> (дата обращения: 22.05.2024). – Текст: электронный.
14. Павловская, Т. А. Программирование на языке высокого уровня C# : краткий курс / Т. А. Павловская. – Москва : ИНТУИТ, 2016. – 182 с. – URL: <https://znanium.ru/catalog/product/2156687> (дата обращения: 21.05.2024). – Режим доступа: по подписке. – Текст : электронный.
15. Рогов, Е. В. PostgreSQL 15 изнутри / Е. В. Рогов. – Москва : ДМК Пресс, 2023. – 664 с. – URL: <https://znanium.ru/catalog/product/2150531> (дата обращения: 20.05.2024). – Режим доступа: по подписке. – Текст : электронный.
16. Смит, Д. Entity Framework Core в действии : практическое руководство / Д. Смит ; пер. с англ. Д. А. Беликова. – Москва : ДМК Пресс, 2023. – 690 с. – URL: <https://znanium.com/catalog/product/2109485> (дата обращения: 27.05.2024). – Режим доступа: по подписке. – Текст : электронный.
17. Сычев, Ю. Н. Защита информации и информационная безопасность : учебное пособие / Ю. Н. Сычев. – Москва : ИНФРА-М, 2023. – 201 с. – (Высшее образование: Бакалавриат). – URL: <https://znanium.ru/catalog/product/1912987> (дата обращения: 28.05.2024). – Режим доступа: по подписке. – Текст : электронный.
18. Торстейнсон, П. Криптография и безопасность в технологии .NET / П. Торстейнсон, Г. А. Ганеш. – 4-е изд. – Москва : Лаборатория знаний, 2020. – 482 с. – (Программисту). – URL: <https://znanium.com/catalog/product/1201346> (дата обращения: 30.05.2024). – Режим доступа: по подписке. – Текст : электронный.
19. Хортон, А. Разработка веб-приложений в ReactJS : практическое руководство / А. Хортон, Р. Вайс ; пер. с англ. Р. Н. Рагимова. – 2-е изд. – Москва : ДМК Пресс, 2023. – 255 с. – URL: <https://znanium.com/catalog/product/2107174> (дата обращения: 24.05.2024). – Режим доступа: по подписке – Текст : электронный..
20. Язык UML: что это такое и зачем он нужен – Текст : электронный // Skillbox: [сайт]. – 2024. – URL: <https://skillbox.ru/media/code/yazyk-uml-chto-eto-takoe-i-zachem-on-nuzhen/> (дата обращения: 20.05.2024).