ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc169182070)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc169182071)

[1.1. Анализ предметной области 5](#_Toc169182072)

[1.2. Постановка задачи 8](#_Toc169182073)

[2. ПРОЕКТНАЯ ЧАСТЬ 10](#_Toc169182074)

[2.1. Проектирование и реализация базы данных системы 10](#_Toc169182075)

[2.2. Проектирование и реализация пользовательского интерфейса системы 12](#_Toc169182076)

[2.2.1. Структура приложения 14](#_Toc169182077)

[2.2.2. Реализация пользовательских форм 16](#_Toc169182078)

[2.2.3. Разработка запросов 28](#_Toc169182079)

[2.2.4. Разработка отчетов 41](#_Toc169182080)

[ЗАКЛЮЧЕНИЕ 43](#_Toc169182081)

[СПИСОК ЛИТЕРАТУРЫ 44](#_Toc169182082)

[Приложение А. Техническое задание 45](#_Toc169182083)

# ВВЕДЕНИЕ

Экологическая обстановка является одним из наиболее важных факторов, влияющих на качество жизни и здоровье людей. Например: загрязнение воздуха, водное загрязнение, разрушение экосистемы. Загрязнение воздуха является основным причиной многих респираторных заболеваний. Токсичные вещества, такие, как диоксид азота и частицы твердых веществ, являются основными загрязнителями воздуха, которые проникают в организм через дыхательные пути и могут нанести серьезный вред. Водное загрязнение также является серьезной проблемой, которая негативно сказывается на здоровье людей. Бактерии, вирусы и токсичные вещества в воде могут вызывать заболевания пищеварительной системы, гепатиты, дизентерию и другие инфекции. Разрушение экосистемы и вырубка лесов приводят к уменьшению биоразнообразия и нарушению природных равновесий. Это может сказаться на возникновении и распространении определенных болезней, которые могут перейти с животных на человека. Кроме того, экологическая обстановка напрямую влияет на психическое состояние человека. Жизнь в загрязненных и неблагоприятных условиях может вызывать стресс, депрессию и другие психические расстройства.

Информационная система (ИС) в экологической сфере позволит собирать данные с помощью сенсоров, других технических устройств и открытых опубликованных источников, передают информацию в центральные базы данных для анализа, обработки и построения моделей. На основе полученных данных, экологические ИС предоставляют информацию управленцам, научным исследователям и другим заинтересованным лицам для принятия решений и разработки стратегий в области экологии и устойчивого развития. ИС также позволяют автоматизировать многие процессы, связанные с экологическим мониторингом, контролем и управлением, что помогает оптимизировать работу и повышать эффективность экологических программ и проектов. Они также способствуют повышению прозрачности и доступности информации для общественности, что стимулирует экологическую осведомленность и активность граждан.

Поэтому создание информационных систем, способных эффективно анализировать данные об экологической ситуации и предоставлять пользователю удобные инструменты для ее улучшения, становится все более актуальной задачей.

Разработка ИС на базе Windows Presentation Foundation (WPF), платформы разработки клиентского ПО от Microsoft, обеспечивает мощные средства для создания современных графических пользовательских интерфейсов и обработки данных. Использование WPF позволяет создавать интуитивно понятные и привлекательные визуальные компоненты, что упрощает работу пользователя и повышает эффективность использования ИС. Для разработки информационных систем на базе Windows Presentation Foundation (WPF) можно использовать Microsoft SQL Server Management Studio (SSMS) для создания, изменения и управления базой данных. WPF предоставляет широкие возможности для создания графического пользовательского интерфейса и работы с данными.

SSMS оснащен набором инструментов для работы с данными, таких как различные окна просмотра данных, редактор запросов, редактор таблиц и другие. Эти инструменты позволяют разработчикам легко работать с данными, отображать их в удобной форме, выполнять запросы и анализировать результаты.

Целью данного курсового проекта является разработка информационной системы, которая будет предоставлять возможности для мониторинга состояния окружающей среды, анализа экологической обстановки и принятия мер по ее улучшению. Она будет базироваться на WPF и будет выполнять такие функции, как сбор данных из различных источников (наблюдений, измерений, статистики), визуализация результатов анализа в удобной форме (графики, диаграммы, карты) и предоставление пользователю инструментов для принятия решений и планирования дальнейших действий.

Разработка такой информационной системы позволит эффективно управлять экологической обстановкой, анализировать ее состояние, выявлять причины нарушений и принимать меры по их устранению. Это также позволит повысить осведомленность и ответственность общества в отношении экологической проблематики и принять участие в ее решении.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* 1. Провести анализ предметной области.
  2. Разработать техническое задание на информационную систему.
  3. Спроектировать БД системы.
  4. Разработать запросы в БД.
  5. Реализовать БД.
  6. Реализовать пользовательский интерфейс.
  7. Разработать пособие и инструкцию по работе с данной информационной системой.
  8. Провести тестирование разработанной информационной системы.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

Учет экологической обстановки является важным аспектом в современном обществе, поскольку охрана окружающей среды стала приоритетным направлением для многих стран, организаций и индивидуалов. Это связано с растущим осознанием людьми важности сохранения природных ресурсов, биологического разнообразия и снижения негативного влияния на климат.

В предметной области учета экологической обстановки рассматриваются следующие аспекты:

1. Измерение и мониторинг загрязнений: Учет экологической обстановки включает системы измерения и мониторинга загрязнений воздуха, воды и почвы. Это позволяет обнаружить и изучить источники загрязнений, определить их влияние на окружающую среду и принять меры для их снижения.
2. Оценка и прогнозирование экологических последствий: Учет экологической обстановки также включает оценку и прогнозирование экологических последствий различных деятельностей, таких как строительство, производство или использование природных ресурсов. Это позволяет принимать решения, основанные на данных о возможных негативных воздействиях на окружающую среду.
3. Управление природными ресурсами: Учет экологической обстановки помогает в управлении природными ресурсами, такими как леса, водные и сельскохозяйственные угодья. Мониторинг состояния и использования этих ресурсов позволяет определить уровень устойчивого их использования и разработать стратегии для их сохранения.
4. Социально-экономический анализ: Учет экологической обстановки также включает социально-экономический анализ влияния экологических факторов на общество и экономику. Это позволяет оценить стоимость потерь, связанных с загрязнением окружающей среды, и определить экономические выгоды от внедрения мер по охране окружающей среды.
5. Развитие экологической политики: Учет экологической обстановки помогает разработке и реализации экологической политики на различных уровнях - от государственных до муниципальных. Он предоставляет информацию о состоянии окружающей среды, ее изменениях и проблемах, требующих вмешательства, и способствует разработке мер по снижению негативного влияния на природу.
6. Осведомленность и образование: Учет экологической обстановки играет важную роль в формировании осведомленности и образования общества в области охраны окружающей среды. Предоставление гражданам доступа к информации о состоянии окружающей среды и ее влиянии на здоровье и благополучие способствует принятию эколог

База данных для исследования экологической обстановки представляет собой набор инструментов, методов и подходов, которые используются для изучения и оценки различных аспектов окружающей среды и ее состояния.

Важным компонентом базы данных является сбор и анализ данных о различных показателях экологической обстановки. Это может включать измерение уровня загрязнения воздуха, воды и почвы, исследование биологического разнообразия, оценку состояния экосистем и многое другое. Для этого используются различные методы, такие как выборочное наблюдение, случайные пробы, эксперименты и моделирование.

База данных также включает в себя разработку стандартов и нормативов, которые помогают сравнить и оценить уровень экологической обстановки в разных регионах и странах. Эти стандарты могут определять допустимые уровни загрязнения, методы оценки экологического воздействия, требования к экологической сертификации и т.д.

Сведения, которые находятся в методической базе - экологическая обстановка:

* Информация о состоянии окружающей природной среды, включая данные о загрязнении воздуха, водных ресурсов, почв, лесов и других экосистем.
* Результаты экологических мониторингов и исследований, проводимых организациями и экспертами в области экологии.
* Статистические данные о производственных и бытовых отходах, их объеме, составе и способах утилизации.
* Информация о химических веществах, воздействующих на окружающую среду, и их влиянии на здоровье человека и экосистемы.
* Методы и приборы для измерения параметров экологической обстановки, такие как уровень шума, концентрация вредных веществ, уровень радиации и другие параметры жизнедеятельности природных и техногенных экосистем.
* Рекомендации по улучшению экологической обстановки и снижению негативного влияния человеческой деятельности на окружающую среду.
* Нормативные документы, законы и правила, регламентирующие охрану окружающей среды и управление природными ресурсами.
* Методология проведения экологической оценки предприятий, территорий и проектов с учетом их воздействия на экологическую обстановку.
* Интернациональные и национальные соглашения и протоколы, касающиеся охраны окружающей среды и снижения выбросов парниковых газов.
* Материалы семинаров, конференций, презентаций и других мероприятий, посвященных экологической проблематике.

Внедрение автоматизации в процессы учета событий имеет множество экологических преимуществ. Устраняя необходимость в физических папках для хранения информации о мероприятиях, мы сокращаем потребление бумаги и помогаем экономить природные ресурсы. Кроме того, цифровая доступность гарантирует, что несколько человек могут одновременно получить доступ к необходимому событию, сводя к минимуму риск дублирования и ненужного использования ресурсов.

Для более детального рассмотрения процесса учета мероприятий рассмотрим рисунок 1 и 2, представленный в виде контекстной диаграммы и диаграммы детализации IDEF0, выполненный в программе Visio.

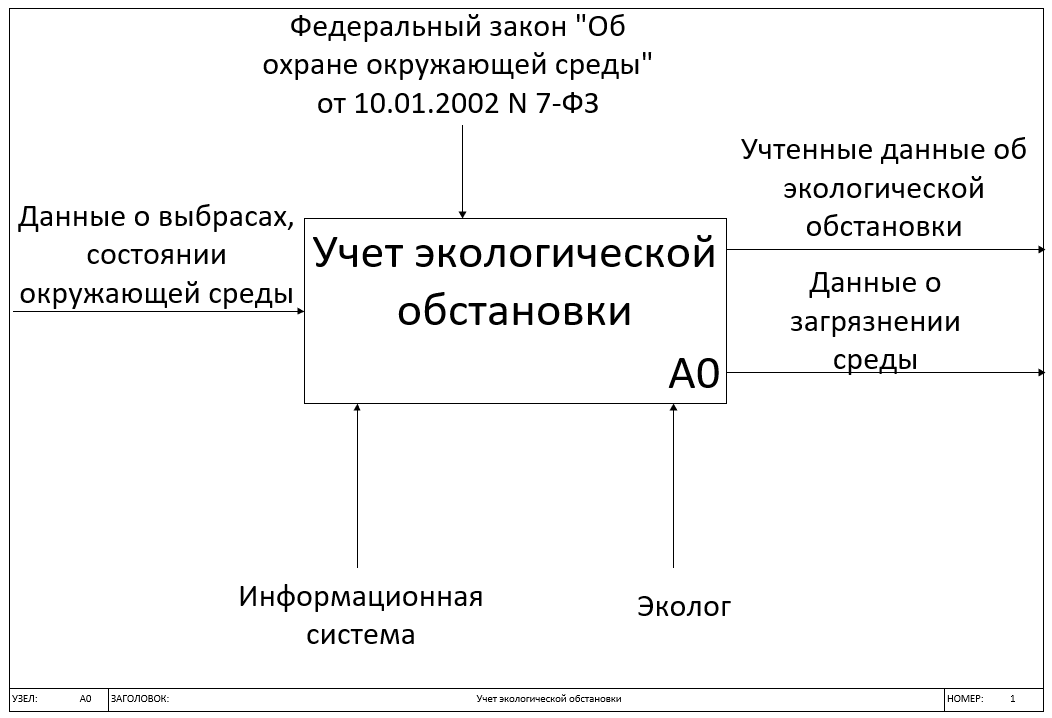


Рисунок 1 – Функциональная диаграмма бизнес-процесса учета экологической обстановки

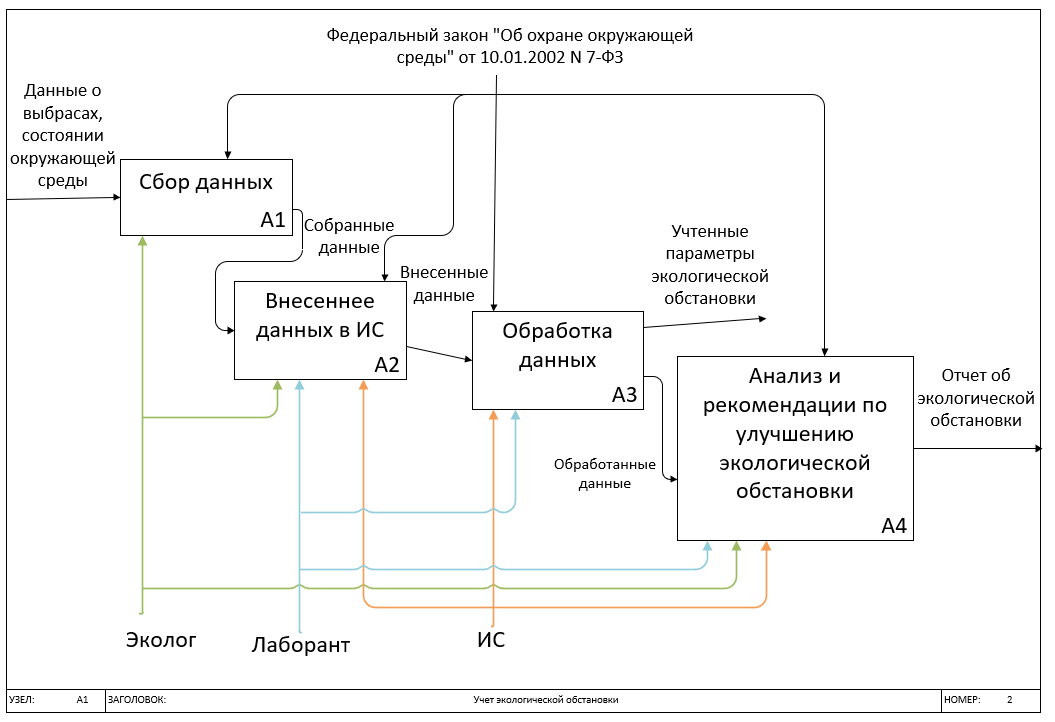


Рисунок 2 – Функциональная диаграмма детализации бизнес-процесса учета экологической обстановки

IDEF0 - Function Modeling - методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов.

Visio — программа для построения диаграмм и векторной графики, разработанная Microsoft. Он позволяет пользователям создавать различные типы диаграмм, такие как блок-схемы, организационные диаграммы и сетевые диаграммы, что делает его полезным для визуализации и передачи сложной информации.

## Постановка задачи

Необходимо создать приложение на платформе WPF с использованием MS SQL сервера и картой для мониторинга экологической обстановки.

Основная цель приложения - предоставить пользователям информацию о степени экологической загрязненности в определенных районах и позволить им вносить свои наблюдения по состоянию окружающей среды.

Основные функциональные требования:

1. Авторизация и регистрация пользователей.
2. Отображение карты.
3. Просмотр данных об экологической обстановке.
4. Добавление наблюдений пользователями.
5. Система уведомлений.
6. Аналитика и статистика.

Подробные требования к разрабатываемой ИС представлены в Техническом задании (см. Приложение А).

# ПРОЕКТНАЯ ЧАСТЬ

## Проектирование и реализация базы данных системы

Разработку базы данных было принято начать с построения ER-модели. ER-модель - [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), позволяющая описывать [концептуальные схемы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%86%D0%B5%D0%BF%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%85%D0%B5%D0%BC%D0%B0) [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C). ER-модель используется при высокоуровневом (концептуальном) [проектировании баз данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Все сущности имеют между собой связь один-ко-многим. ER-модель представлена на

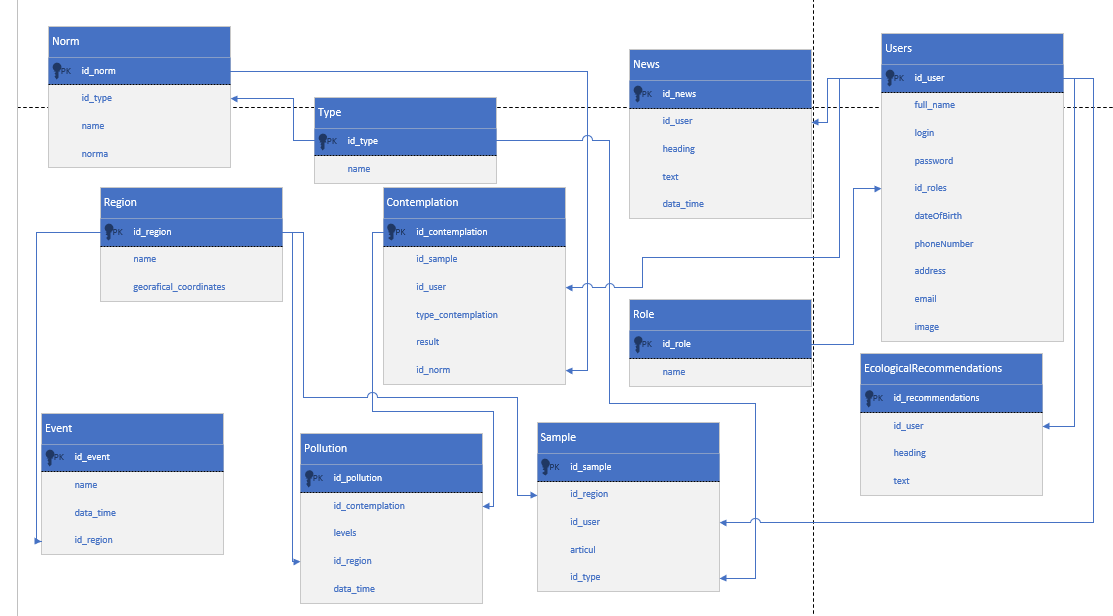


Рисунок 3 – ER-модель БД

Разработанная ER-модель состоит из 11 сущностей: Пользователи, Новости, Роли, Экологические мероприятия, Изучения проб, Пробы, Тип, Норма, Регион, Мероприятие, Загрязнение.

Сущности: Роли, тип обследования, норма, созданы для подстановки в сущности: Пробы, Изучения проб и роли в сущность пользователь.

* Сущности: Пользователи, загрязнение, образец, организации, новости, регион, мероприятие, являются основными рабочими сущностями в БД. Каждая сущность необходима для своей задачи: Загрязнение: представляет различные типы загрязнений, уровень загрязнения, источники загрязнения, время.
* Образец: Содержит данные об объекте обследования или анализа в контексте экологии
* Тип обследования: Описывает различные виды экологических обследовании.
* Норма: Содержит установленные стандарты, нормативы и пороговые значения для экологических показателей, которые определяют допустимые уровни загрязнения и качество окружающей среды.
* Регион: представляет информацию о различных регионах, их географии, природных ресурсах, их состоянии.
* Мероприятие: содержит информацию о различных планируемых или проводимых экологических и природоохранных мероприятиях.
* Новости: содержит новостную информацию об экологических событиях, проектах и других новостях в области окружающей среды.
* Роли: сущность, описывающая различные роли или категории пользователей, которые могут иметь различные разрешения и функции в системе.
* Пользователь, содержит в себе информацию о пользователях системы.

После проектирования ER-модели был выполнен экспорт в MS SQL - server. Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL, создан совместно Microsoft и Sybase. Transact - SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. В MS SQL - server были установлены типы данных для каждой таблицы.

На рисунке 3 представлена физическая схема БД, реализованная на платформе MS SQL Server 2018. На модели отражены названия атрибутов и их типы данных на сервере. Так же на модели отображены связи между сущностями для обеспечения целостности данных.

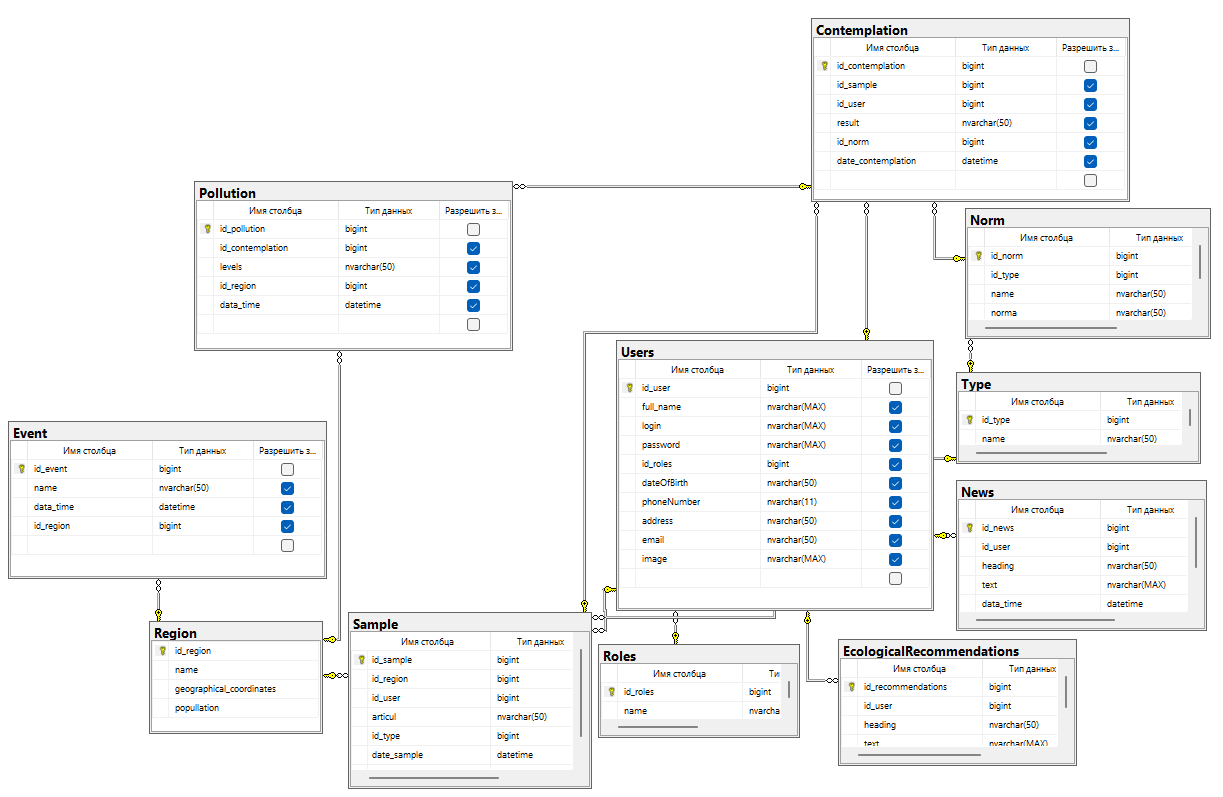
****

Рисунок 4 – Физическая модель БД

## Проектирование и реализация пользовательского интерфейса системы

Для реализации пользовательского интерфейса была выбрана интегрированная среда разработки Visual Studio 2019.

Visual Studio - это популярная интегрированная среда разработки (IDE) от компании Microsoft, предназначенная для создания различных типов приложений, включая настольные приложения, веб-приложения, мобильные приложения, игры и облачные решения. Visual Studio включает в себя широкий спектр инструментов, редактор кода, отладочные средства, средства тестирования, поддержку версионирования и другие функции, которые делают процесс разработки визуальным и удобным. Visual Studio также обладает богатой экосистемой расширений и интеграцией с другими сервисами и платформами разработки, что делает ее популярным инструментом среди разработчиков. Данные продукты позволяют, разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии Windows Presentation Foundation, а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone) [.NET Compact Framework](https://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight).

Система программирования Visual Studio версии 2019 фирмы Microsoft предоставляет наиболее широкие возможности для программирования приложений ОС Windows.

Преимущества Visual Studio по сравнению с аналогичными программными продуктами это:

* Обширные возможности: Visual Studio обеспечивает широкий спектр инструментов и функций для разработки программного обеспечения, включая поддержку различных языков программирования, инструменты для тестирования и отладки, а также возможности для создания веб-приложений, мобильных приложений, облачных решений и других типов приложений.
* Интеграция: Visual Studio интегрируется с другими популярными инструментами и платформами разработки, такими как Azure, GitHub, Docker, и многими другими, что обеспечивает беспрепятственную работу на различных этапах разработки.
* Поддержка платформ: Visual Studio обеспечивает инструменты для разработки под различные платформы, включая Windows, Android, iOS, Linux, и многое другое, что делает его удобным выбором для многоплатформенной разработки.
* Поддержка языков: Visual Studio поддерживает широкий спектр языков программирования, включая C#, C++, JavaScript, Python, и другие, что делает его универсальным инструментом для разработчиков различных специализаций.

Так же был использован язык программирования C# так как:

1. Универсальность: C# является мощным и универсальным языком программирования, который поддерживает разработку разнообразных типов приложений, включая настольные приложения, веб-приложения, мобильные приложения и облачные решения.
2. Интеграция с платформой .NET: C# разработан специально для платформы .NET, что обеспечивает широкие возможности для интеграции с другими технологиями, библиотеками и инструментами, предоставляемыми .NET.
3. Безопасность и надежность: C# предлагает высокий уровень безопасности и проверки типов, что делает его более надежным по сравнению с некоторыми другими языками.
4. Модернизация: C# активно развивается и модернизируется командой разработчиков Microsoft, включая новые функции, улучшения производительности и поддержку современных технологий.
5. Общий стандарт: C# имеет множество общих стандартов, синтаксических схожести с другими популярными языками программирования, такими как Java или C++, что упрощает кросс-платформенную разработку и переход других разработчиков на C#.

* Xamarin. Благодаря покупке Xamarin на C # теперь можно писать под Android и iOS. Это, конечно, большой плюс, так как их собственная мобильная ОС (Windows Phone) не завоевала большой популярности;
* добавлено функциональное программирование (F #);
* большое сообщество программистов;
* много вакансий на должность C # программиста в любом регионе.

### Структура приложения

Для реализации поставленных задач была выбрана клиент-серверная архитектура системы мониторинга экологической обстановки.

1. Клиентская часть
   1. Собирает данные с датчиков и устройств мониторинга.
   2. Передает собранные данные на сервер.
   3. Предоставляет пользователям интерфейс для просмотра и анализа данных.
2. Серверная часть
   1. Хранит и управляет данными, собранными с датчиков.
   2. Обрабатывает и анализирует данные для выявления закономерностей и тенденций.
   3. Предоставляет пользователям через веб-интерфейс или API доступ к данным и результатам анализа.
3. Преимущества клиент-серверной архитектуры
   1. Масштабируемость: Архитектура позволяет легко добавлять новые клиенты и серверы для обработки растущего объема данных.
   2. Доступность: Данные хранятся на централизованном сервере, что обеспечивает их доступность для всех авторизованных пользователей.
   3. Безопасность: Контроль доступа и защита данных могут быть централизованы на сервере, что повышает безопасность системы.
   4. Эффективность: Вычисления и обработка данных выполняются на сервере, что освобождает клиентские устройства от необходимости выполнять сложные задачи.
   5. Обслуживаемость: Обновления и обслуживание системы могут выполняться централизованно на сервере, что упрощает управление.
4. Дополнительные преимущества для системы мониторинга экологической обстановки:
   1. Надежность: Данные хранятся на отказоустойчивых серверах, что обеспечивает их сохранность и доступность даже в случае сбоев оборудования.
   2. Анализ в реальном времени: Сервер может обрабатывать данные в режиме реального времени, позволяя пользователям оперативно реагировать на изменения экологической обстановки.
   3. Интеграция: Архитектура позволяет легко интегрировать систему с другими системами управления данными и аналитическими инструментами.

Сервер БД

Клиент

Рисунок 5 – Схема клиент-серверного приложения

Разработанное клиентское приложение выполняет все поставленные задачи, и имеет сложную архитектуру, представленную на Рисунке 6.

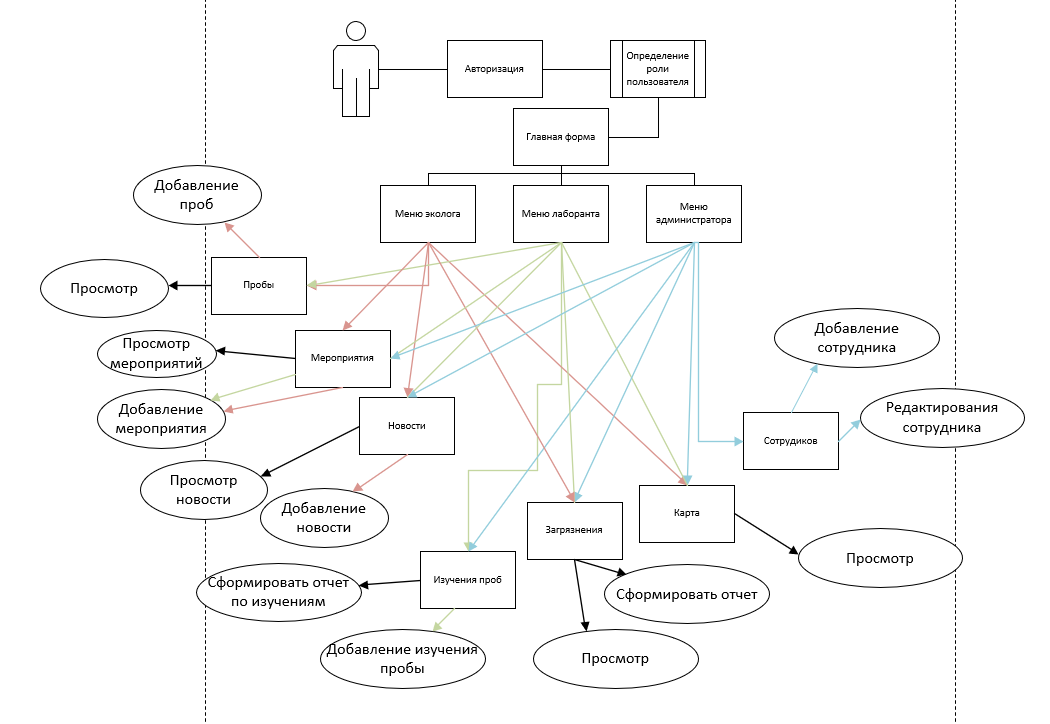


Рисунок 6 – Схема приложения

На схеме разными цветами выделены разграничения прав доступа. Красным цветом отмечены функции доступные Экологам, зеленым цветом отмечены функции доступные лаборантам, желтым цветом отмечены функции доступные администраторам, черным выделены общие функции для всех групп.

### Реализация пользовательских форм

В ходе курсового проекта с помощью программного продукта Microsoft Visual Studio было создано приложение. Так же с помощью SQL Server Management Studio была создана база данных, которая была подключена к этому приложению.

Первым окном было разработано окно приветствия и загрузки приложения, потом окна регистрации, авторизации и сброса пароля при помощи OTP кода на почту. На Рисунке 7 и 8 представлены окна загрузки и о приложении. На Рисунке 9 и 10 представлена форма авторизации.

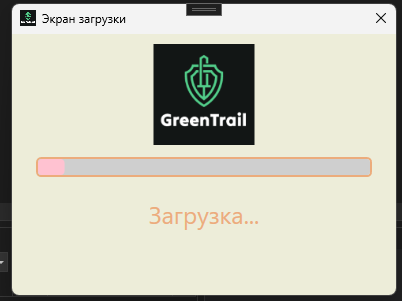


Рисунок 7 – Окно загрузки

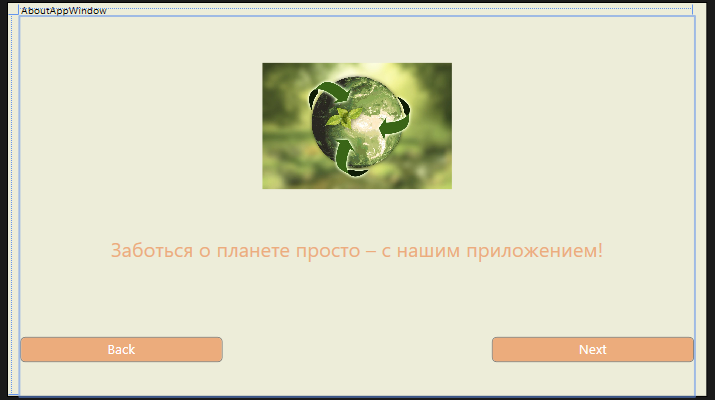


Рисунок 8 – Окно приветствия

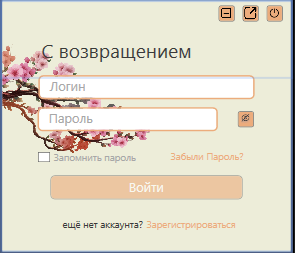


Рисунок 9 – Окно авторизации светлая тема

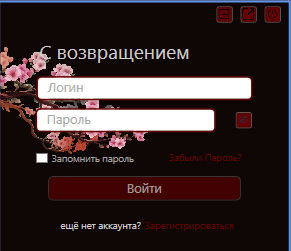
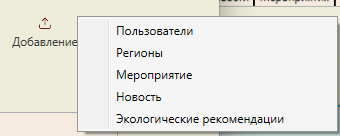
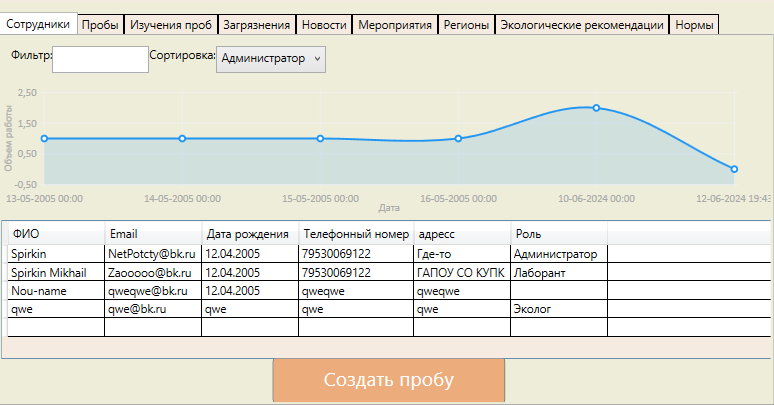


Рисунок 10 – Окно авторизации темная тема

Дальнейшие картинки будут показывать только светлую тему

Авторизоваться можно с 1 лаборанта, 1 эколога и 1 админа. У эколога ограниченный доступ к функция добавления изучения проб и пользователей. Пример ограничений доступом можно рассмотреть на рисунках 8 Возможности Администратора и 9 Возможности эколога, и 10 возможности Лаборанта



Рисунки 11 – Возможности администратора.

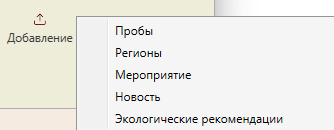
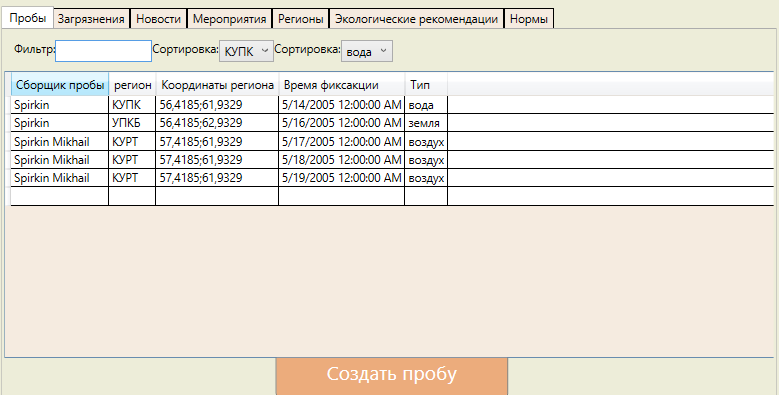


Рисунок 12 – Возможности эколога.

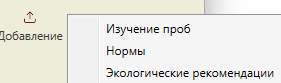
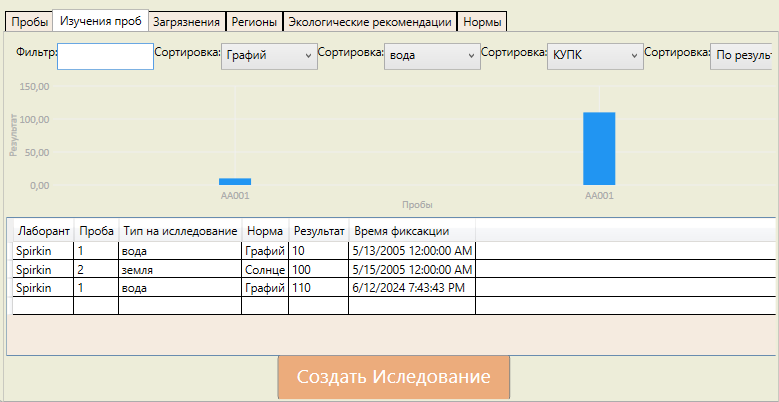
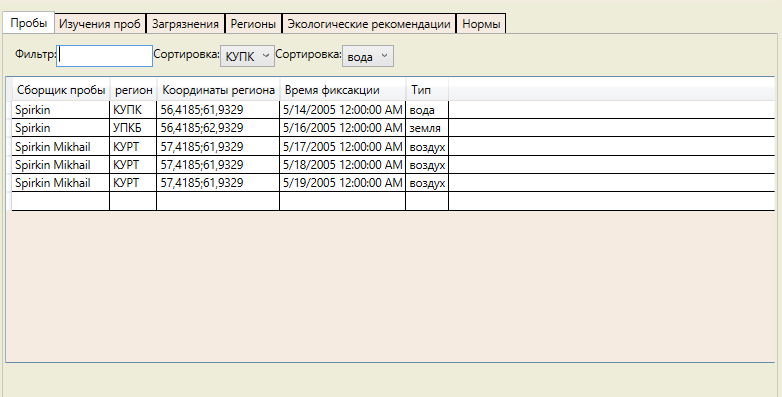


Рисунок 13 – Возможности лаборанта.

Код авторизации:

private void btn\_login\_Click(object sender, RoutedEventArgs e)

{

string hashedPassword;

if (\_isShowingPassword)

{

hashedPassword = Funs.ComputeSHA256Hash(tb\_pass.Text);

DataBaseFuns.AuthenticateUser(tb\_login.Text, hashedPassword, this);

}

else

{

hashedPassword = Funs.ComputeSHA256Hash(pb\_pass.Password);

DataBaseFuns.AuthenticateUser(tb\_login.Text, hashedPassword, this);

}

if (cb\_rememberMe.IsChecked.Value)

{// Сохраняем флаг показа в реестр

RegistryKey key = Registry.CurrentUser.OpenSubKey("Software\\GreenTrail", true);

key.SetValue("RememberMe", cb\_rememberMe.IsChecked.Value);}}

public static void AuthenticateUser(string username, string password,Window window)

{

try

{

Users user = GreanTrailEntities.GetContext().Users.FirstOrDefault(u => u.login == username && u.password == password);

if (user != null)

{// запомнить ID пользователя

Funs.RememberUserId(user.id\_user);

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

window.Close();

}

else

{

MessageBox.Show("Упс, ваш аккаунт не найден!", "Упс!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Что-то пошло не так(",MessageBoxButton.OK,MessageBoxImage.Error );

}

}

После авторизации мы переходим к главной форме на которой происходит работа пользователя или администратора с приложением. На главной форме представлено: самописный ToolBar сверху, картинка профиля, если она есть и красивая картинка по теме, далее идет меню: ”Просмотр”, “Карта”, “Экспорт”, “Добавление”. После меню идут блоки: “Новости”, “Мероприятия”, “Экологические рекомендации”.

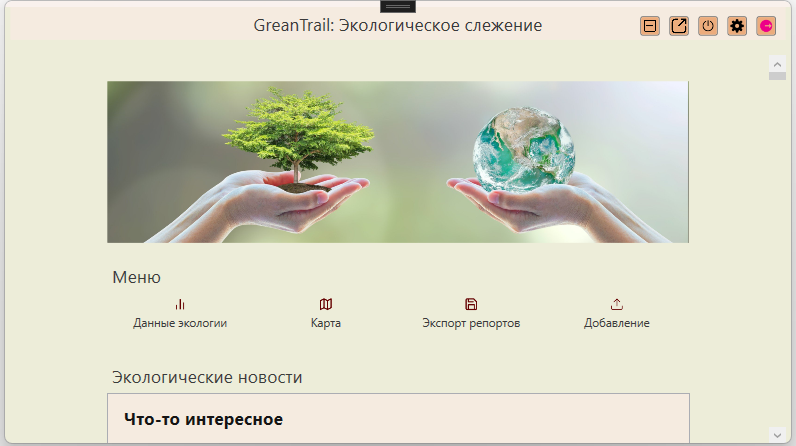


Рисунок 14 – Главная форма.

Переход на формы реализован:

«Название формы» «название переменной» = new «Название формы»();

«название переменной».Show();

this.Close();

Так же для отображения Новостей, мероприятий и экологических рекомендаций были разработаны ModelView

private void LoadData()

{

using (var dbContext = new GreanTrailEntities())

{

var news = dbContext.News.ToList();

foreach (var item in news)

{NewsViewModels.Add(new NewsViewModel

{

Id = item.id\_news,

Title = item.heading,

Text = item.text,

Autor = item.Users.full\_name,

Date = (DateTime)item.data\_time

});

}

//добавление в ViewModel Event и EcologicalRecommendations }}

public class NewsViewModel

{

public long Id { get; set; }

public string Title { get; set; }

public string Text { get; set; }

public string Autor { get; set; }

public DateTime Date { get; set; }}

WPF — современный и гибкий графический интерфейс пользователя (GUI) для приложений .NET. Он позволяет разработчикам создавать привлекательные и интерактивные пользовательские интерфейсы с использованием привязки данных, стилизации и анимации.

Возможности WPF:

* Поддержка различных устройств, включая сенсорные экраны и планшеты
* Высокая производительность и эффективность
* Декларативное программирование, упрощающее определение визуальной части интерфейса
* Экспорт пользовательских интерфейсов в различные форматы (например, XAML, PDF)
  + - 1. Библиотеки для расширения WPF:
         1. Entity Framework: Библиотека для работы с данными, позволяющая разработчикам использовать объекты .NET для моделирования и взаимодействия с реляционными базами данных.
         2. Microsoft Toolkit: Коллекция библиотек для расширения функциональности WPF, включая элементы управления, утилиты и службы.
         3. LiveChart: Библиотека для создания интерактивных и настраиваемых диаграмм и графиков в WPF-приложениях.
         4. Microsoft.Maps.MapControl.WPF: Библиотека для создания и отображения карт в приложениях Windows Presentation Foundation (WPF). Она позволяет разработчикам легко добавлять карты Bing и другие картографические сервисы в свои приложения.
      2. Библиотеки для экспорта в различные файлы:
         1. ClosedXML: Библиотека для работы с электронными таблицами Excel, позволяющая разработчикам создавать, редактировать и экспортировать файлы Excel из WPF-приложений.
         2. Spire.XLS: Библиотека для работы с электронными таблицами, поддерживающая широкий спектр форматов файлов, таких как Excel, CSV и HTML.
         3. iTextSharp: Библиотека с открытым исходным кодом для создания и управления документами PDF в приложениях .NET. Она предоставляет широкий набор функций для создания сложных и настраиваемых PDF-документов.
         4. OfficeOpenXml: Библиотека с открытым исходным кодом для работы с документами Microsoft Office (Word, Excel, PowerPoint) в приложениях .NET. Она позволяет разработчикам создавать, редактировать и управлять документами Office без необходимости использования самого пакета Microsoft Office.

### Разработка запросов

В данном приложение есть 17 запросов:

* В окне «Настройки»:
* Запрос на смену темы;
* Запрос на смену пароля;
* Запрос на загрузку картинки;
* Запрос на изменения текущего авторизованного пользователя.

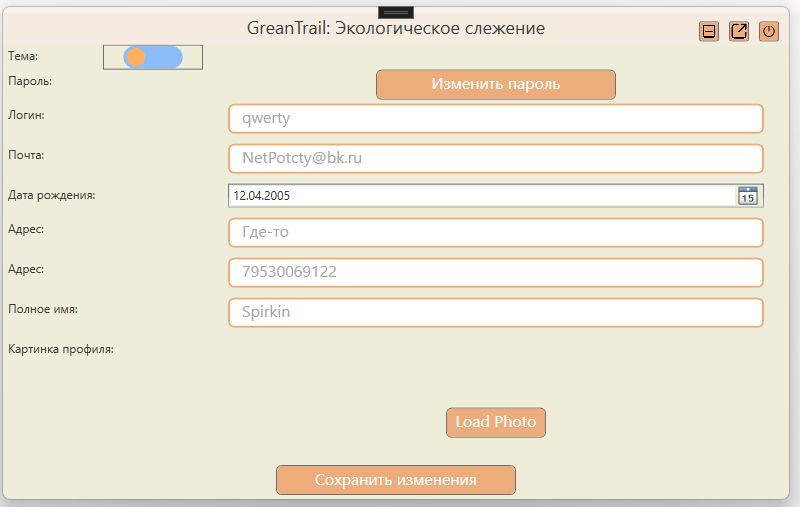


Рисунок 15 Окно настроек.

* В Окне «Добавления»:
* Запросы на подстановку значений;
* Запрос на добавление выбранной сущности;

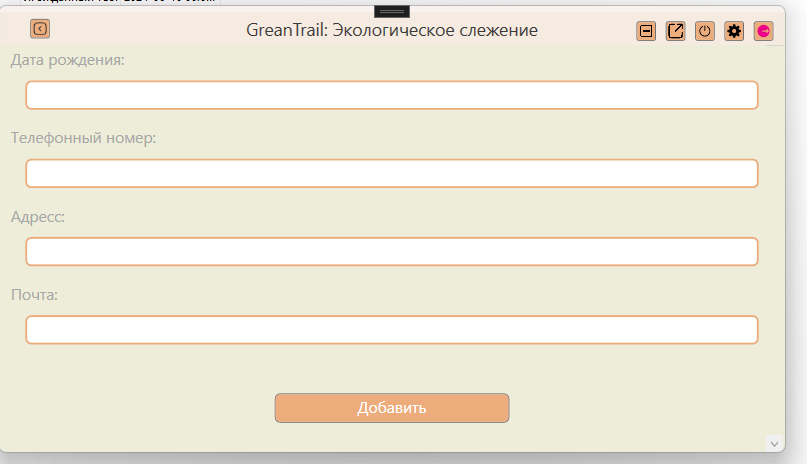


Рисунок 16 – Окно добавления пользователя.

Код добавления:

private void AddButton\_Click(object sender, RoutedEventArgs e)

{if (areAllFieldsFilled)

{

// Получение выбранной сущности

string selectedEntity = table;

var currentUser = DataBaseFuns.GetCurrentUser();

switch (selectedEntity)

{case "Образец":

if (SampleArticulTextBox.Text != string.Empty && SampleRegionComboBox.SelectedItem != null && SampleTypeComboBox.SelectedItem != null)

{

Sample sample = new Sample

{id\_region = (int)SampleRegionComboBox.SelectedValue + 1,

id\_user = currentUser.id\_user, // Взять из текущего пользователя

articul = SampleArticulTextBox.Text,

id\_type = (int)SampleTypeComboBox.SelectedValue + 1,

date\_sample = DateTime.Now

};

\_context.Sample.Add(sample);

}

else { MessageBox.Show("Вы не ввели все данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;

case "Изучение пробы":

if (ContemplationSampleComboBox.SelectedItem != null && ContemplationResultTextBox.Text != string.Empty && ContemplationNormComboBox.SelectedItem != null)

{ Contemplation contemplation = new Contemplation

{ id\_sample = (int)ContemplationSampleComboBox.SelectedIndex + 1,

id\_user = currentUser.id\_user, // Взять из текущего пользователя

result = ContemplationResultTextBox.Text, // Временно задано произвольное значение

id\_norm = (int)ContemplationNormComboBox.SelectedIndex + 1,

date\_contemplation = DateTime.Now};

Sample sample = greanTrailEntities.Sample.FirstOrDefault(r => r.articul == (string)ContemplationSampleComboBox.SelectedItem);

Norm norma = greanTrailEntities.Norm.FirstOrDefault(r => r.id\_norm == (long)ContemplationNormComboBox.SelectedIndex + 1 && r.id\_type == sample.id\_type);

Region region = greanTrailEntities.Region.FirstOrDefault(r => r.id\_region == sample.id\_region);

if (Convert.ToInt32(norma.norma) < Convert.ToInt32(ContemplationResultTextBox.Text))

{

Pollution pollution = new Pollution

{ id\_contemplation = contemplation.id\_contemplation,

levels = (Convert.ToInt32(ContemplationResultTextBox.Text) - Convert.ToInt32(norma.norma)).ToString(),

id\_region = region.id\_region,

data\_time = DateTime.Now,};

\_context.Pollution.Add(pollution);}

\_context.Contemplation.Add(contemplation);}

else { MessageBox.Show("Вы не ввели все данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;

case "Пользователи":

if (UserLoginTextBox.Text != string.Empty && UserLoginTextBox.Text != string.Empty)

{ Users users = new Users

{

full\_name = UserFullNameTextBox.Text,

login = UserLoginTextBox.Text,

password = Funs.ComputeSHA256Hash(UserPassTextBox.Text),

id\_roles = (int)UserRolesComboBox.SelectedIndex + 1,

dateOfBirth = UserDateOfBirthTextBox.Text,

phoneNumber = UserPhoneNumberTextBox.Text,

address = UserAddressTextBox.Text,

email = UserEmailTextBox.Text,};

\_context.Users.Add(users);}

else { MessageBox.Show("Вы не ввели все обязательные данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;

case "Мероприятие":

if (UserLoginTextBox.Text != string.Empty && UserLoginTextBox.Text != string.Empty)

{Event events = new Event

{name = tblock\_heading.Text,

data\_time = Calendar.SelectedDate};

\_context.Event.Add(events);}

else { MessageBox.Show("Вы не ввели все обязательные данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;

case "Новость":

if (UserLoginTextBox.Text != string.Empty && UserLoginTextBox.Text != string.Empty)

{News news = new News

{id\_user = currentUser.id\_user,

heading = tblock\_heading.Text,

text = TextBox\_text.Text,

data\_time = DateTime.Now,};

\_context.News.Add(news);}

else { MessageBox.Show("Вы не ввели все обязательные данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;

case "Рекомендация":

if (UserLoginTextBox.Text != string.Empty && UserLoginTextBox.Text != string.Empty)

{EcologicalRecommendations ecologicalRecommendations = new EcologicalRecommendations

{id\_user = currentUser.id\_user,

heading = tblock\_heading.Text,

text = TextBox\_text.Text,

};\_context.EcologicalRecommendations.Add(ecologicalRecommendations);}else { MessageBox.Show("Вы не ввели все обязательные данные!", "Erorr", MessageBoxButton.OK, MessageBoxImage.Information); return; }

break;}

\_context.SaveChanges();

// Вывод сообщения об успешном добавлении

MessageBox.Show("Данные успешно добавлены в базу данных.");}}

* В окне «Экспорт»:
  + Запросы на выбор таблицы;
  + Запрос на выбор формата;
  + Запрос на выбор “сохранить” или “Сохранить как”;

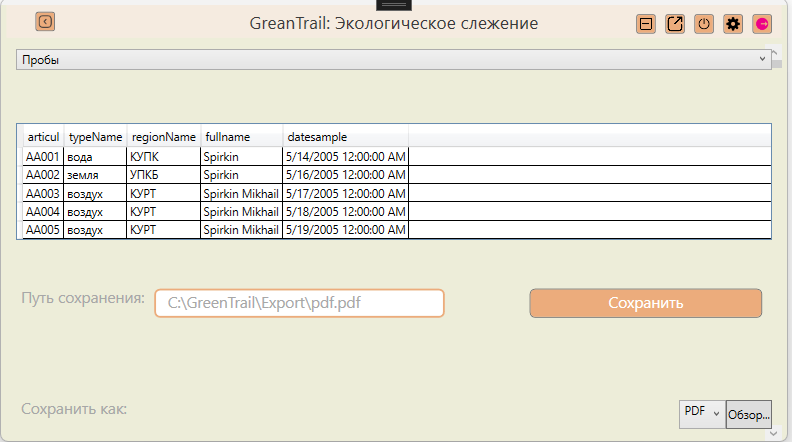


Рисунок 16 –Окно экспорта.

Код экспортирования на примере PDF:

public void Save(string filePath, DataGrid dataGrid)

{

// Сохранить данные DataGrid в выбранном формате

switch (\_selectedFormat.ToLower())

{

case "pdf":

SaveToPdf(filePath, dataGrid);  
private void SaveToPdf(string filePath, DataGrid dataGrid)

{

try

{

var data = dataGrid.ItemsSource.Cast<object>().ToList();

// Создать объект Document для генерации PDF

var document = new iTextSharp.text.Document(iTextSharp.text.PageSize.A4, 25, 25, 25, 25);

var writer = PdfWriter.GetInstance(document, new FileStream(filePath, FileMode.Create));

document.Open();

// Добавить заголовки столбцов

var table = new PdfPTable(dataGrid.Columns.Count);

foreach (var column in dataGrid.Columns)

{

table.AddCell(new PdfPCell(new Phrase(column.Header.ToString())));

}

// Добавить данные строк

foreach (var item in data)

{

var properties = item.GetType().GetProperties();

foreach (var property in properties)

{

object value = property.GetValue(item);

string stringValue = value != null ? value.ToString() : string.Empty;

table.AddCell(new PdfPCell(new Phrase(stringValue)));

}

}

// Добавить таблицу в документ

document.Add(table);

// Закрыть документ

document.Close();

}catch (Exception ex)

{MessageBox.Show(ex.Message,"Упс",MessageBoxButton.OK,MessageBoxImage.Error);}}

* В окне «Карта»:
  + Запросы на отображения меток;
  + Запрос на получении информации о метке;

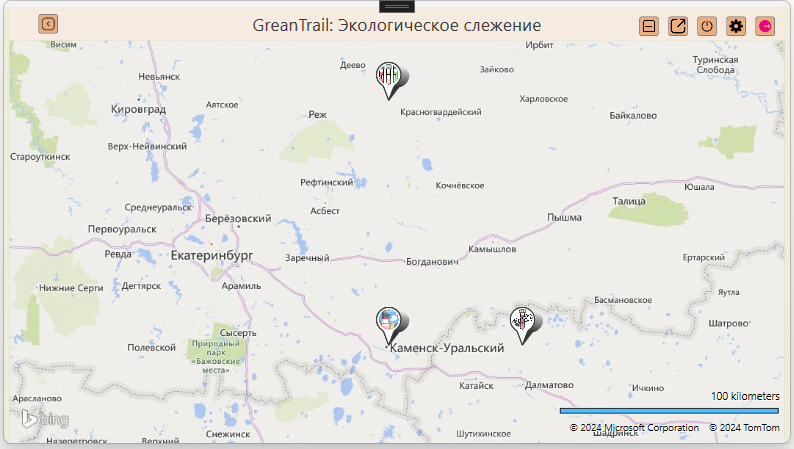


Рисунок 17 –Окно карты.

Верстка карты:  
<m:Map Grid.Row="1" x:Name="myMap" Mode="Road" ZoomLevel="10" Center="56.4185,61.9329">

<m:Map.CredentialsProvider>

<m:ApplicationIdCredentialsProvider ApplicationId="1UmrVJDK1N97FZ6Kq9kK~sEN4QKO9OaUTOwuSdKyR1Q~Ah8XCDz1Txx7VwqXG-k0fJ6kItJ8tyOfEPwKkYLjBU8Lt4lH9-q3iwTd6-fYQADi" />

</m:Map.CredentialsProvider>

</m:Map>

* В окне «Просмотр»:
  + Запросы на отображения таблиц по роли;
  + Запрос на переход в добавление по по роли;
  + Запрос на построение графиков;
  + Запрос на получении информации по графику;
  + Запрос на сортировку;
  + Запрос на фильтрацию;

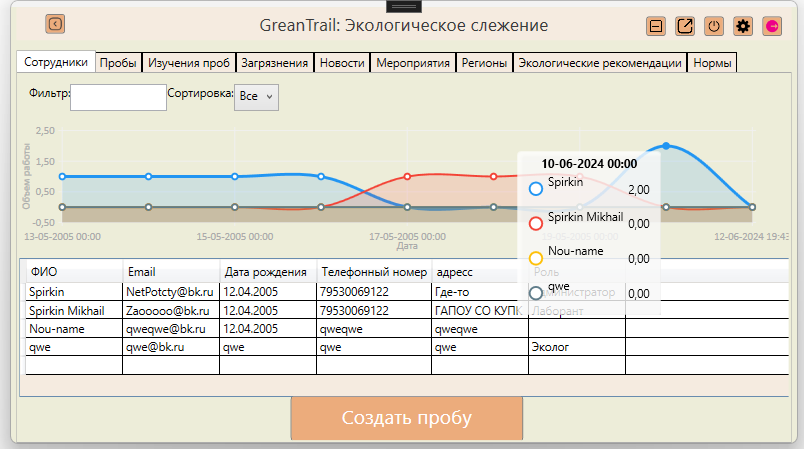


Рисунок 22 –Окно просмотра.

### Разработка отчетов

Для разработки отчетов были использованы различные библиотеки для того чтоб выводить данные в форматах: "PDF", "JSON", "TXT", "DOC", "DOCX", "XLS", "XLSX", "CSV".

* PDF:

iTextSharp: Библиотека с открытым исходным кодом для создания и управления документами PDF.

* JSON:

Newtonsoft.Json: Популярная библиотека для сериализации и десериализации объектов JSON.

* TXT:

System.IO: Стандартная библиотека .NET для работы с файлами, позволяющая экспортировать таблицы в текстовые файлы с использованием класса `StreamWriter`.

* DOC/DOCX:

ClosedXML: Библиотека для работы с электронными таблицами, которая поддерживает экспорт таблиц в форматы DOC и DOCX.

* XLS/XLSX:

Spire.XLS: Коммерческая библиотека для работы с электронными таблицами, которая поддерживает экспорт таблиц в форматы XLS и XLSX.

* CSV:

ClosedXML: Библиотека для работы с электронными таблицами, которая также позволяет экспортировать таблицы в файлы CSV.

В данном приложении имеются 3 отчета:

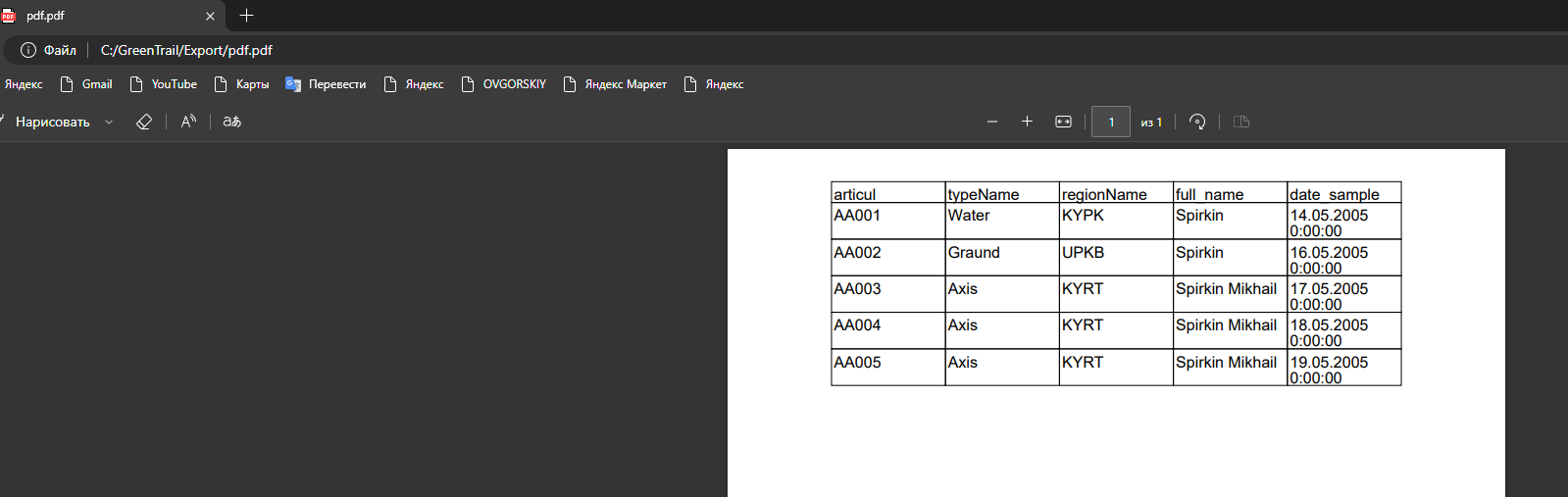


Рисунок 16 - отчет о собранных пробах

# ЗАКЛЮЧЕНИЕ

Цели поставленные в начале разработки были выполнены полностью, приложение имеет удобный и простой интерфейс, а также выполняет все необходимые действия.

Разработанное приложение может быть использовано в open-source проектах связанных с экологической обстановкой.

Целью данной информационной системы является своевременное обеспечение надлежащих людей надлежащей [информацией](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), то есть для удовлетворения конкретных информационных потребностей в рамках определенной [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является информационная продукция - документы, [информационные массивы](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2&action=edit&redlink=1), [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и информационные услуги, и снижение рутиной работы с бумагами во время подбора мероприятий.

Во время выполнения курсового проекта были использованы следующие программные продукты:

* Visual Studio 2022
* MS SQL Server
* SQL Server Management Studio

Подробно изучил работу с библиотеками:

* LinQ
* Entity FrameWork
* Newtonsoft.Json
* iTextSharp
* ClosedXML
* LiveCharts
* Microsoft.Maps.MapControl.WPF
* OfficeOpenXml

Приложение может:

* формировать документы с таблицами;
* осуществлять учет экологической обстановки путем сбора проб их анализом и расположения на карте;
* менять тему;
* добавлять данные проб, их исследованием, регионом, новостей, рекомендаций, мероприятий связанных.

Выполняя данный курсовой проект, была более углубленно изучена работа с Visual Studio 2022 с MS SQL Server.

# СПИСОК ЛИТЕРАТУРЫ

1. Голицына, О.Л. и др. Базы данных; Форум; Инфра-М, 2013
2. Емельянова Н.З. Основы построения автоматизированных информационных систем. – М., Форум, 2007
3. Олифер В.Г. Компьютерные сети. Принципы, технологии. - СПб., Питер, 2009
4. Баженова И. Ю. SQL и процедурно-ориентированные языки. - Национальный Открытый Университет «ИНТУИТ», 2016 г.
5. Гвоздева В.А. Основы построения автоматизированных информационных систем. – М., Форум, 2009
6. Дунаев В..В. Базы данных Язык SQL. – СПб., БХВ - Петербург, 2006
7. Максимов Н.В., Попов И.И. Компьютерные сети.- М, ФОРУМ: ИНФРА-М, 2010
8. Малыхина М.П. Базы данных: основы, проектирования, использование. – СПб., БХВ-Петербург, 2006
9. Мельников В.П. Информационная безопасность. – М., Академия, 2009
10. Лазицкас Е. А., Загумённикова И. Н., Гилевский П. Г. Базы данных и системы управления базами данных: учебное пособие. – РИПО, 2016 г.
11. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 - 6-е издание, 2012
12. Документация Microsoft по C#: https://docs.microsoft.com/en-us/dotnet/csharp/
13. Документация Microsoft по WPF: https://docs.microsoft.com/en-us/dotnet/desktop/wpf/
14. Документация Microsoft по LINQ: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/
15. Документация Microsoft по Entity Framework: https://docs.microsoft.com/en-us/ef/core/
16. Документация по Newtonsoft.Json: https://www.newtonsoft.com/json
17. Документация по iTextSharp: https://lowagie.com/iTextSharp/
18. Документация по ClosedXML: https://github.com/ClosedXML/ClosedXML
19. Документация по LiveCharts: https://lvcharts.net/
20. Документация по Microsoft.Maps.MapControl.WPF: https://docs.microsoft.com/en-us/dotnet/api/microsoft.maps.mapcontrol.wpf.map?view=mapcontrol-wpf
21. Документация по OfficeOpenXml: <https://github.com/closedxml/officeopenxml>

## Приложение А. Техническое задание