Частное учреждение образования

«Колледж бизнеса и права»

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ АНАЛИЗА

МЕСТОРАПОЛОЖЕНИЯ ЖИВОТНЫХ В НАЦИОНАЛЬНОМ ПАРКЕ

«БЕЛОВЕЖСКАЯ ПУЩА»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по учебному предмету «Базы данных и системы

управления базами данных»

КП.Т091004.401

Руководитель проекта ( Н.В.Ржеутская )

Обучающийся ( А.О.Зеневич )

2024

СОДЕРЖАНИЕ

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*3*

*КП Т.091004.401 ПЗ*

*Разраб.*

*Зеневич А.О.*

*Провер.*

*Ржеутская Н.В.*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Разработка программного средства для анализа*

*месторасположения*

*животных в Национальном*

*парке «Беловежская пуща»*

*Лит.*

*Листов*

*КБП*

*у*

[Введение 3](#_Toc161609165)

[1 Описание задачи 5](#_Toc161609166)

[1.1 Анализ предметной области 5](#_Toc161609167)

[1.2 Постановка задачи 5](#_Toc161609168)

[2 Проектирование системы 7](#_Toc161609169)

[2.1 Требования к приложению 7](#_Toc161609170)

[2.2 Проектирование модели 7](#_Toc161609171)

[2.3 Проектирование структуры базы данных 8](#_Toc161609172)

[2.4 Концептуальный прототип 10](#_Toc161609173)

[3 Описание реализации программного средства 16](#_Toc161609174)

[3.1 Инструменты разработки и применяемые технологии 16](#_Toc161609175)

[3.2 Порядок авторизации пользователей 17](#_Toc161609176)

[3.3 Организация данных 18](#_Toc161609177)

[3.4 Функции: логическая и физическая организация 21](#_Toc161609178)

[3.5 Входные и выходные данные 24](#_Toc161609179)

[3.6 Функциональное тестирование 24](#_Toc161609180)

[3.7 Описание справочной системы 26](#_Toc161609181)

[4 Применение 28](#_Toc161609182)

[4.1 Назначение программного средства 28](#_Toc161609183)

[4.2 Условия применения 28](#_Toc161609184)

[Заключение 29](#_Toc161609185)

[Список информационных источников 30](#_Toc161609186)

[Приложение А Текст программных модулей 31](#_Toc161609187)

[Приложение Б Результаты работы программы 32](#_Toc161609188)

# Введение

В настоящее время развитие информационных технологий оказывает существенное влияние на различные сферы жизни человека, в том числе и на животноводство. Современные технологии позволяют создавать программные средства, способные оптимизировать процессы управления и контроля за животным миром, а также предоставлять эффективные инструменты для анализа местоположения животных.

В рамках данного курсового проекта ставится задача разработки программного средства, который будет предоставлять возможность просмотра информации о животных, их типах, месторасположении и локациях, посещенных ими. Реализуемое программное средство позволит осуществлять поиск, создание, изменение и удаление данных о животных, их типах, точках местоположения и местах посещения животных.

Актуальность поставленной задачи обусловлена необходимостью создания эффективных инструментов для управления и контроля за животным миром, а также необходимостью автоматизации процессов сбора и обработки информации, связанной с животными. Кроме того, разработка подобных сервисов способствует развитию индустрии информационных технологий и расширению возможностей их применения в различных сферах деятельности.

К курсовому проекту также прилагается пояснительная записка, состоящая из четырех разделов и графической части.

В первом разделе в «Описании задачи» приводится описание предметной области, включающий в себя описание бизнес-процессов, происходящих в области управления и контроля за животным миром, описание алгоритмов получения промежуточных и итоговых показателей, а также обоснование необходимости компьютерной обработки информации или процессов. Определяется потенциальная аудитория, которой будет предоставлен доступ к сервису.

Во второй главе «Проектирование системы» проводится постановка требования к приложению, описываются накладываемые ограничения, общие требования к интерфейсу, способы защиты и сохранности данных, а также концептуальные прототипы.

В третьей главе «Описание реализации программного средства» приводятся инструменты разработки и применяемые технологии, порядок авторизации пользователей, описание физической организации данных в контексте среды разработки, описываются используемые функции и процедуры, входные и выходные данные, описание справочной системы, а также приводится функциональное тестирование на основании тест-кейсов и скриншотов.

В четвёртом разделе «Применение» описывается назначение программного средства, цели разработки, саму установку программного средства с описанием всего дерева файлов, описывается область применения, а также класс решаемых задач. Приводятся ограничения, накладываемые на область применения.

В заключении будет проанализирована поставленная задача, определена степень соответствия поставленной задачи и выполненной работы, а также преимущества и недостатки программного средства.

Приложение А будет содержать текст модулей программного средства.

Приложение Б представлены результаты работы программного средства.

Графическая часть представлена диаграммами сущность-связь (концептуальная и логическая модель), вариантов использования, диаграмма деятельности.

# Описание задачи

## Анализ предметной области

В разрабатываемом программном средстве точки текущего местоположения животных будут заданы: импортироваться из БД.

В данном разделе рассмотрим предметную область проекта, то есть перемещение животных, а также бизнес-процессы, происходящие в этой области.

Перемещение животных является важным аспектом. Оно может быть вызвано различными причинами, такими как поиск еды, воды или укрытия, а также в связи с процессом размножения. Кроме того, перемещение животных может быть связано с миграцией, изменением климатических условий или просто с природным инстинктом животного.

Для эффективного учёта за перемещениями животных необходима компьютерная обработка информации об их передвижении. Поэтому, в рамках данного курсового проекта, будет решена задача разработки программного средства для отслеживания перемещения.

При проектировании данного программного средства необходимо учитывать потенциальную аудиторию, которая может использовать его в своей работе. В первую очередь это могут быть зоопарки, ветеринарные клиники, рыбхозы, птицефабрики и т.д.

## Постановка задачи

Первоочередной функцией программного средства является просмотр информации о животном. Пользователь должен иметь возможность получать информацию о конкретном животном, включая его тип, текущее местоположение, а также все места, которые он посетил, а также поиск животного по его текущему местоположению, а также по всем местам, которые он посетил.

Далее следует разработка функции поиска, создания, изменения и удаления животного для администратора. В случае изменения информации о животном, администратор должен иметь возможность обновить информацию о его типе, местоположении или посещенных местах. При необходимости администратор сможет удалить запись о животном.

Функционал программного средства включает также точки местоположения, которые представляют собой географические координаты мест, где были замечены животные. Пользователь может просмотреть информацию о точке локации, а администратор создать новую точку, изменить ее параметры или удалить ее из системы.

Наконец, программное средство должно предоставлять функции по созданию, изменению и удалению места посещения животных. Пользователь сможет просмотреть информацию о перемещении животного, а администратор новую запись о посещенном месте, изменить ее параметры или удалить ее.

При проектировании разрабатываемого программного средства необходимо провести анализ существующих аналогов с указанием отличий, которые будут реализованы в разрабатываемом сервисе.

В данное время отсутствуют полноценные аналоги разрабатываемого программного средства. Существует один платный веб-сайт, который предоставляет сходный функционал. Однако, поскольку доступ к этому сайту требует платы, невозможно провести полный анализ его плюсов и минусов. Одним из негативных аспектов такого аналога является то, что он предоставляется за плату, что может быть недоступно или нежелательно для пользователей, ищущих бесплатные решения.

Разрабатываемое программное средство будет обладать всем вышеописанным функционалом, предоставляться в свободном доступе.

# Проектирование системы

## Требования к приложению

Стилистическое оформление приложения должно соответствовать техническому заданию.

[Требования к графическому дизайну](#_Toc87949487):

* дизайн программного средства должен быть лаконичным (создание меню, кнопочных форм, панелей инструментов) и в то же время выглядеть стильно, современно;

Требования к шрифтовому оформлению:

* размер (кегль) шрифтов должен обеспечивать удобство восприятия текста при минимально допустимом размере экрана.

Система управления контентом программного средства должна обеспечить возможность выполнения следующих действий:

Администратору:

* осуществить возможность ведение базы данных о типах животных, животных, локациях, точках перемещения животных.

Пользователю:

* осуществить возможность просмотра информации о типах животных, животных, локациях, точках перемещения животных;
* поиск информации по различным критериям (по локации (животное и дата посещения), по животному (история перемещения животного), по точке на карте локации);
* экспорт данных в Excel и XML.

## Проектирование модели

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций.

Разработанные диаграммы приводятся в графической части на соответствующих листах. После описания делается ссылка на соответствующий лист в графической части.

Разработаны следующие диаграммы:

* вариантов использования с указанием типов связей между вариантами использования. Для основных прецедентов по тексту пояснительной записки необходимо составить сценарии;
* деятельности.

Диаграмма вариантов использования описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования. Она является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества так называемых вариантов использования, предоставляемых системой множеству актеров или сущностей, взаимодействующих с системой. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером.

Варианты использования определяют функциональные возможности. Каждый из них представляет определенный способ использования. Таким образом, каждый вариант использования соответствует последовательности действий для того, чтобы клиент мог получить определенный результат.

Исходя из диаграммы вариантов использования можно выделить следующие функции как:

* разделение ролей;
* работа с таблицей «Животное»;
* работа с таблицей «Типы животных»;
* работа с таблицей «Локация»;
* работа с таблицей «Точка перемещения»;
* поиск.

Диаграмма вариантов использования представлена в графической части на листе 3.

Диаграмма деятельности представляет собой UML-диаграмму, на которой показаны действия, состояния которых описаны на диаграммах состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграмма деятельности представлена в графической части на листе 4.

## Проектирование структуры базы данных

Главной целью проектирования модели является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования данных является диаграмма «Сущность-связь» (ERD), которые предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обозначений для определения данных и отношения между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных и абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности.

Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения типа «отец-сын» или производственные отношения типа «начальник-подчиненный». Другой тип связей задается отношениями «иметь в собственности» или «обладать свойством».

Графическая модель строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей.

Исходя из предметной области можно выделить следующие сущности разработки: «Пользователи», «Типы животных», «Животные», «Локации», «Точки перемещения».

Для сущности «Пользователи» атрибутами будут являться:

* никнейм;
* пароль;
* имя;
* возраст;
* электронная почта.

Для сущности «Типы животных» атрибутами будут являться:

* наименование типа;
* описание типа.

Для сущности «Животные» атрибутами будут являться:

* наименование животного;
* описание животного.

Для сущности «Локации» атрибутами будут являться:

* наименование локации;
* описание локации;
* широта;
* долгота.

Для сущности «Точки перемещения» атрибутами будут являться:

* дата.

Диаграмма «Сущность-связь» представлена на рисунке 2.1

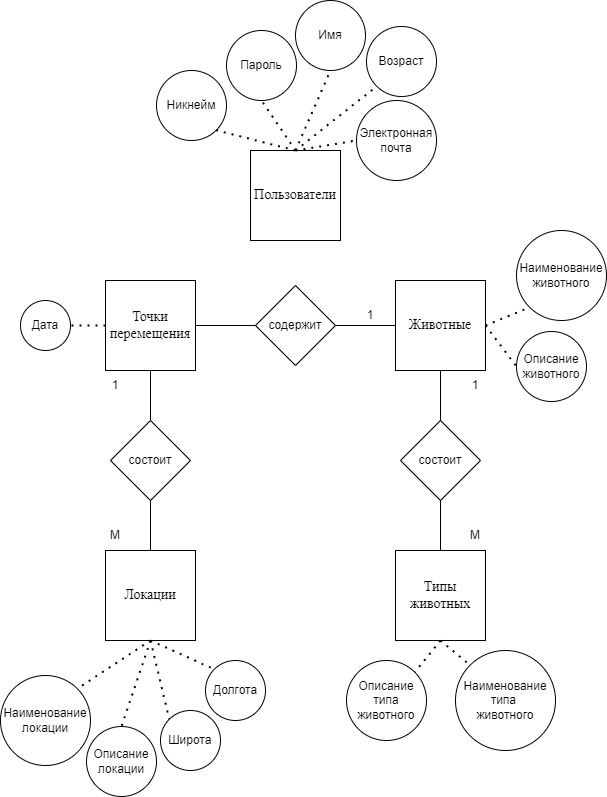


Рисунок 2.1 – Диаграмма «Сущность-связь»

## Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным средством работы с пользователем. Разрабатываемое приложение будет содержать несколько форм.

Все формы будут содержать стандартные пользовательские элементы управления.

В рабочем режиме программы, администратору и пользователю, для удобной навигации, будет предоставлено меню.

На главной странице расположен весь необходимый для администратора и пользователя функционал.

Слева располагается меню навигации, где находятся всё доступные для анализа месторасположения животных в Национальном парке «Беловежская пуща».

Администратор может добавить тип животного, животное, локацию или точку перемещения путём нажатия на кнопку «Добавить». Макет страниц для добавления тип животного, животное, локацию или точку перемещения представлен рисунке 2.2-2.5.

Кнопка «Типы животных» позволяет администратору добавлять данные о новых типах животных или изменять уже существующие. Представлено на рисунке 2.2.

Кнопка «Животные» позволяет администратору добавлять данные о новых животных или изменять уже существующие. Представлено на рисунке 2.3.

Кнопка «Локации» позволяет администратору добавлять данные о новых локациях или изменять уже существующие. Представлено на рисунке 2.4.

Кнопка «Точки перемещения» позволяет администратору добавлять данные о новых точках перемещения или изменять уже существующие. Представлено на рисунке 2.5.

Все страницы главной формы будут иметь схожий интерфейс. Макет главной формы для администратора с открытой формой отображения статистики представлены на рисунке 2.6.

Пользователь имеет возможность просматривать формы (таблицы) «Тип животного», «Животные», «Локации», «Точки перемещения», а также фильтровать данные в них по заданному критерию. Для экспорта данных в Excel или Xml, пользователь может нажать на кнопки «Экспорт в Excel» или «Экспорт в Xml». Макет страниц для данного функционала представлен на рисунках 2.7-3.0.

Также пользователь может осуществить поиск информации по различным критериям (по локации (животное и дата посещения), по животному (история перемещения животного), по точке на карте локации). Макет представлен на рисунке 3.1.

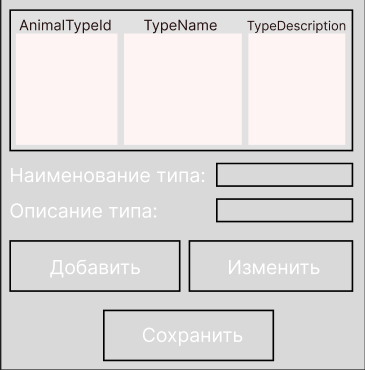


Рисунок 2.2 – Макет формы «Типы животных»

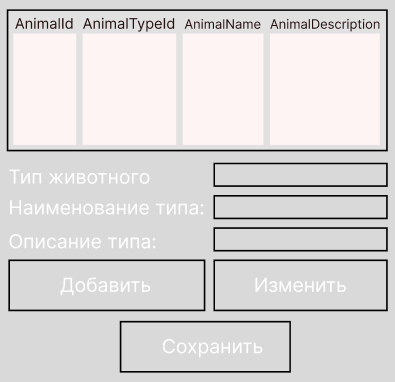


Рисунок 2.3 – Макет формы «Животные»

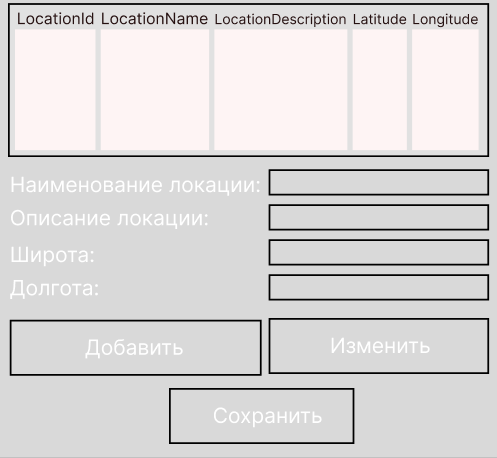


Рисунок 2.4 – Макет формы «Локации»



Рисунок 2.5 – Макет формы «Точки перемещения»

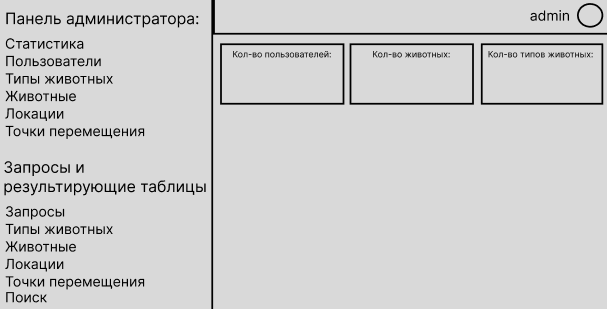


Рисунок 2.6 – Макет начального меню администратора

с открытой формой «Статистика»

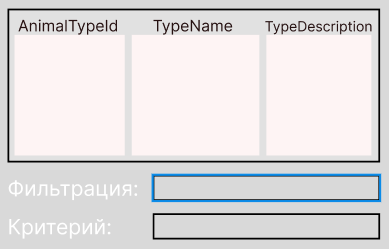


Рисунок 2.7 – Пользовательский макет формы «Типы животных»

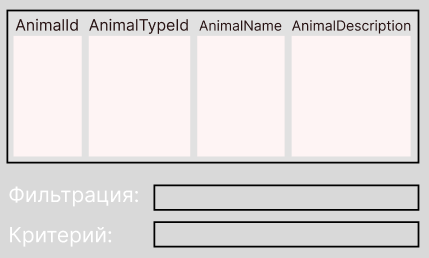


Рисунок 2.8 – Пользовательская форма «Животные»

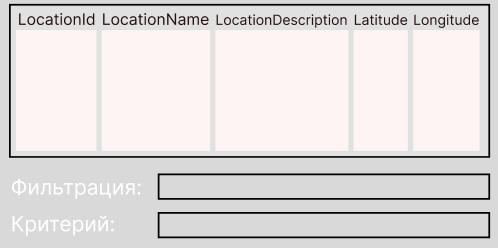


Рисунок 2.9 – Пользовательская форма «Локации»



Рисунок 3.0 – Пользовательская форма «Точки перемещения»

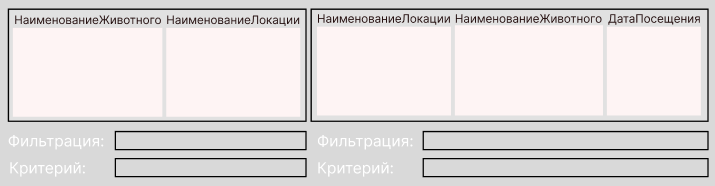


Рисунок 3.1 – Пользовательская форма «Поиск»

# Описание реализации программного средства

## Инструменты разработки и применяемые технологии

Ниже описаны инструменты разработки, которые будут использоваться для написания программного средства.

Ноутбук Asus Rog Strix G15 G513RM-LN007 со следующей аппаратной конфигурацией:

* процессор AMD Ryzen 7-6800H, 2022 года выпуска, техпроцесс 6 нанометров, частота 4.7 ГГц, 8 ядер, 16 потоков;
* оперативная память DDR5 16 Гб, тактовая частота 4800 МГц;
* видеокарта Nvidia GeForce RTX 3060 8 Гб (версия драйвера 551.76);
* встроенная видеокарта AMD Radeon (TM) Graphics;
* винчестер SSD 1024 Гб.

Программное обеспечение:

* операционная система Windows 11 (версия 23H2, сборка OC 22631.3296);
* среда программирования Visual Studio 2022 с использованием Guna.UI2 Framework (версия 2.0.4.6);
* язык программирования C# (версия 7.3);
* реляционная база данных Microsoft SQL Server (версия: 16.0);
* интегрированная среда Microsoft SQL Server Management Studio 19 (версия 19.3.4);
* офисный пакет приложений Microsoft Office (MS Office) 2021 (версия 2402, сборка 17328.20162);
* программа для построения диаграмм Draw.io.

Операционная система – это набор управляющих программ, предназначенных для управления ресурсами вычислительной системы как единого комплекса, другими словами, операционная система – это набор программного обеспечения, который обеспечивает работу компьютера.

В ходе выполнения проекта были задействованы следующие подключаемые модули приложения:

Windows 11 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

Microsoft Visual Studio ­– это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений, это также линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов [10].

Язык C# выбран в качестве основного преимущественно из-за того, что он предоставляет большие возможности для разработчиков при создании многопоточных и асинхронных приложений, что необходимо в современных web-приложениях во время обработки сложных запросов, для обеспечения плавности интерфейса [3].

Guna.UI2 – это фреймворк пользовательского интерфейса, который создает компьютерные клиентские приложения. Платформа разработки WinForms поддерживает широкий набор функций разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязку данных, документы и безопасность [8].

СУБД Microsoft SQL Server 2019 – основной продукт Microsoft по обработке данных. Версия 2018 – это наиболее значительный прорыв в истории платформ данных Microsoft: более быстрые транзакции и запросы, ценные сведения на любом устройстве, расширенная аналитика, новые технологии безопасности и новые варианты использования при применении гибридного облака. SQL Server 2019 предоставляет передовые функции для решения задач высокой важности за счет выполнения операций в памяти и встроенной операционной аналитики [11].

SQL Server – это хорошо масштабируемый, полностью реляционный, быстродействующий многопользовательский сервер баз данных масштаба предприятия, способный обрабатывать большие объемы данных для клиент-серверных приложений.

SQL Server Management Studio (SSMS) – это интегрированная среда для управления любой инфраструктурой SQL: от SQL Server до базы данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL. С помощью SSMS можно развертывать, отслеживать и обновлять компоненты уровня данных, используемые вашими приложениями, а также создавать запросы и скрипты.

.NET Framework – [программная платформа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0#%D0%BA%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%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), выпущенная компанией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в [2002 году](https://ru.wikipedia.org/wiki/2002_%D0%B3%D0%BE%D0%B4). Основой платформы является общеязыковая среда исполнения [Common Language Runtime (CLR)](https://ru.wikipedia.org/wiki/Common_Language_Runtime), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду [10].

Microsoft Office 2021 – офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows.

Dr.Explain – программа для быстрого создания и редактирования пользовательской и справочной документации.

Draw.io – используется для создания таких типов диаграмм, как блок-схемы, Организационные диаграммы, Поэтажные планы, Сетевые диаграммы, Диаграммы UML, Интеллектуальные карты и многое другое. Он также обычно используется для таких сценариев, как отображение процессов и визуальная совместная работа.

## Порядок авторизации пользователей

Всего в программном средстве есть две роли:

* администратор;
* пользователь.

Их данные хранятся в таблице «Пользователи» и содержит следующие атрибуты:

* уникальный идентификатор;
* никнейм;
* пароль;
* возраст;
* электронная почта.

Осуществляется вход всех пользователей в одном окне программы, а затем уже с помощью внутренних алгоритмов вычисляется с какой ролью пользователь входит в программу, после этого уже открывается окно с соответствующими возможностями, исходя из привилегий той или иной роли.

Роль «Администратор» обладает следующими привилегиями:

* просмотр статистики;
* администрирование пользователей (изменение и удаление пользователей, выдача ролей);
* все те же возможности, что и у роли «Пользователь».

Роль «Пользователь» обладает следующими привилегиями:

* просмотр формы «Типы животных» с функционалом фильтрации и сортировки по критериям;
* просмотр формы «Животные» с функционалом фильтрации и сортировки по критериям;
* просмотр формы «Локации» с функционалом фильтрации и сортировки по критериям;
* просмотр формы «Точки перемещения» с функционалом фильтрации и сортировки по критериям;
* просмотр формы «Поиск» с функционалом фильтрации и сортировки по критериям.

## Организация данных

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы – атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

База данных соответствует реляционной модели данных, где каждой выделенной в ходе проектирования сущности соответствует таблица.

В таблицах 3.1-3.5 представлена организация данных в базе данных «Microsoft SQL Server», а также краткое описание.

Таблица «Пользователь» хранит информацию о пользователях, структура которой представлена в таблице 3.1.

Таблица 3.1 – Таблица «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничение | Ключ |
| Id | INT | n N:(n > 0) | Первичный |
| Никнейм | NVARCHAR (50) | NOT NULL |  |
| Пароль | NVARCHAR (50) | NOT NULL |  |
| Имя | NVARCHAR (50) | NOT NULL |  |
| Возраст | NVARCHAR (50) |  |  |
| Электронная почта | NVARCHAR (50) |  |  |

Таблица «Типы животных» хранит информацию о типах животных, структура которой представлена в таблице 3.2.

Таблица 3.2 – Таблица «Типы животных»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничение | Ключ |
| Id типа  животного | INT | n N:(n > 0) | Первичный |
| Имя типа  животного | NVARCHAR (50) | NOT NULL |  |
| Описание типа  животного | NVARCHAR (50) | NOT NULL |  |

Таблица «Животные» хранит информацию о животных, структура которой представлена в таблице 3.3.

Таблица 3.3 – Таблица «Животные»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничение | Ключ |
| Id животного | INT | n N:(n > 0) | Первичный |
| Id типа  животного | INT | n N:(n > 0) | Внешний |
| Имя животного | NVARCHAR (50) | NOT NULL |  |
| Описание  животного | NVARCHAR (255) | NOT NULL |  |

Таблица «Локации» хранит информацию о локациях, структура которой представлена в таблице 3.4.

Таблица 3.4 – Таблица «Локации»

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничение | Ключ |
| Id локации | INT | n N:(n > 0) | Первичный |
| Имя локации | NVARCHAR (50) | NOT NULL |  |
| Описание  локации | NVARCHAR (255) | NOT NULL |  |
| Широта | DECIMAL (9, 6) |  |  |
| Долгота | DECIMAL (9, 6) |  |  |

Таблица «Точки перемещения животного» хранит информацию о точках перемещения животных, структура которой представлена в таблице 3.5.

Таблица 3.5 – Таблица «Точки перемещения животного»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Тип данных | | Ограничение | Ключ |
| Id точки  перемещения животного | INT | | n N:(n > 0) | Первичный |
| ID локации | INT | | n N:(n > 0) | Внешний |
| ID животного | INT | | n N:(n > 0) | Внешний |
| Продолжение таблицы 3.5 | | |  |  |  |
| Название | Тип данных | | Ограничение | Ключ |
| Дата | DATETIME | |  |  |

Физическая схема базы данных представлена на рисунке 3.1.

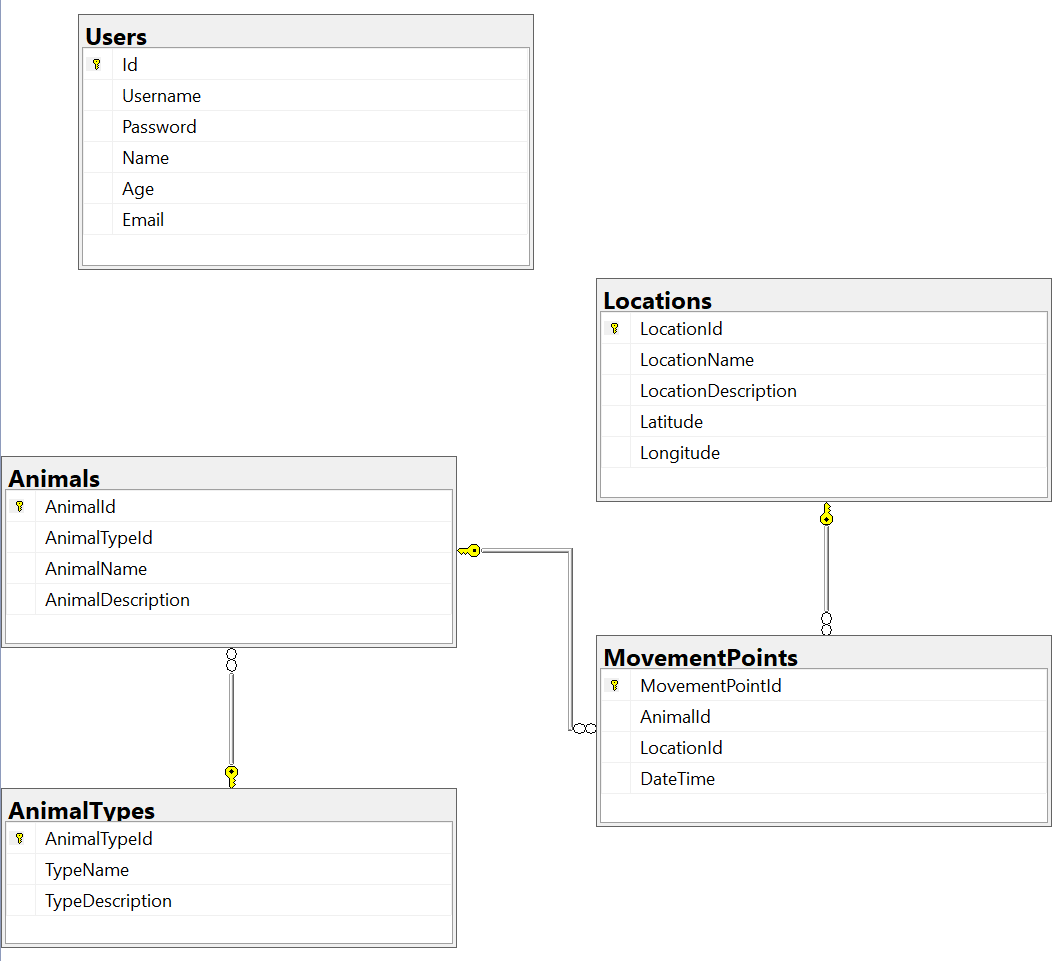


Рисунок 3.1 – Физическая схема базы данных

## Функции: логическая и физическая организация

На основании диаграммы вариантов использования в программном средстве для анализа месторасположения животных в Национальном парке «Беловежская пуща» реализована функция поиска информации по различным критериям (по локации (животные и дата посещения), по животному (история перемещения животного), по точке на карте локации).

Рассмотрим данные функции программы, в интерфейсе программного средства закреплена за элементом управления «SearchUserButton», обрабатывается событием SearchUserButton\_Click (). Код функции представлен ниже.

private void tbFilter\_TextChanged(object sender, EventArgs e)

{

if (cbFilter.SelectedItem == null) return;

var bs = (BindingSource)v4DataGridView.DataSource;

var dv = (DataView)bs.List;

var selectedText = cbFilter.SelectedItem.ToString().ToLower();

if (dv.Table.Columns[selectedText].DataType == typeof(Int32))

{

dv.RowFilter = $"CONVERT({selectedText}, 'System.String') LIKE '%{tbFilter.Text.ToLower()}%'";

}

else

{

dv.RowFilter = $"{selectedText} LIKE '%{tbFilter.Text.ToLower()}%'";

}

}

private void cbFilter\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbFilter.SelectedItem == null) return;

var bs = (BindingSource)v4DataGridView.DataSource;

var dv = (DataView)bs.List;

var selectedText = cbFilter.SelectedItem.ToString().ToLower();

if (dv.Table.Columns[selectedText].DataType == typeof(Int32))

{

dv.RowFilter = $"CONVERT({selectedText}, 'System.String') LIKE '%{tbFilter.Text.ToLower()}%'";

}

else

{

dv.RowFilter = $"{selectedText} LIKE '%{tbFilter.Text.ToLower()}%'";

}

}

Реализована функция экспорта данных в таблицу Excel.

Рассмотрим данные функции программы, в интерфейсе программного средства закреплена за элементом управления «ExportToExcelButton», обрабатывается событием ExportToExcelButton\_Click (). Код функции представлен ниже.

private void ExportToExcel(string fileName)

{

try

{

using (var package = new ExcelPackage())

{

var workSheet = package.Workbook.Worksheets.Add("Животные");

workSheet.Cells["A1"].Value = "IdТипаЖивотного";

workSheet.Cells["B1"].Value = "НаименованиеЖивотного";

workSheet.Cells["C1"].Value = "ОписаниеЖивотного";

workSheet.Cells["A1:C1"].Style.Font.Bold = true;

workSheet.Cells["A1:C1"].Style.HorizontalAlignment = ExcelHorizontalAlignment.Center;

for (int row = 0; row < v2DataGridView.Rows.Count; row++)

{

for (int col = 0; col < v2DataGridView.Columns.Count; col++)

{

workSheet.Cells[row + 2, col + 1].Value = v2DataGridView[col, row].Value;

}

}

workSheet.Cells[workSheet.Dimension.Address].AutoFitColumns();

package.SaveAs(new FileInfo(fileName));

MessageBox.Show("Данные экспортированы в файл " + fileName, "Экспорт завершен", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("Произошла ошибка при экспорте данных: " + ex.Message, "Ошибка экспорта", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ExportToExcelButton\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog = new SaveFileDialog

{

Filter = "Excel Files|\*.xlsx",

Title = "Выберите файл для экспорта данных в Excel"

};

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

ExportToExcel(saveFileDialog.FileName);

}

}

Реализована функция экспорта данных в Xml.

Рассмотрим данные функции программы, в интерфейсе программного средства закреплена за элементом управления «ExportToXmlButton», обрабатывается событием ExportToXmlButton\_Click (). Код функции представлен ниже.

private void ExportToXml(string fileName)

{

try

{

DataSet dataSet = new DataSet();

dataSet.Tables.Add(v2TableAdapter.GetData().Copy());

dataSet.WriteXml(fileName);

MessageBox.Show("Данные экспортированы в файл " + fileName, "Экспорт завершен", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Произошла ошибка при экспорте данных: " + ex.Message, "Ошибка экспорта", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void ExportToXmlButton\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog = new SaveFileDialog

{

Filter = "XML Files|\*.xml",

Title = "Выберите файл для экспорта данных"

};

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

ExportToXml(saveFileDialog.FileName);

}

}

## Входные и выходные данные

Входными являются данные, введённые администратором в таблицы базы данных: «Типы животных», «Животных», «Локации», «Точки перемещения».

Входными данным при добавлении данных о новом типе животного:

* наименование типа животного;
* описание типа животного.

Входными данным при добавлении данных о новом животном:

* внешний ключ типа животного;
* наименование животного;
* описание животного.

Входными данным при добавлении данных о новой локации:

* наименование локации;
* описание локации;
* широта;
* долгота.

Входными данным при добавлении данных о новом типе животного:

* внешний ключ животного;
* внешний ключ локации.

Вся вносимая в программное средство информация хранится в локальной базе данных.

Выходными данными являются результаты поиска данных по различным критериям. Например, при поиске данных по типу животного, выходной информацией будет сформированный список животного с выбранным типом.

А также выходными данными будут функции экспорта данных в Excel и Xml.

## Функциональное тестирование

В процессе написания приложения необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям и проводится для выявления неполадок и недочетов программы на этапе ее сдачи в эксплуатацию.

Тестирование программного средства будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

Тестирование приложения будет оформлено в виде тест-кейсов. Все действия начинаются с главной страницы приложения. В таблицах 3.6-3.8 приводится тест-кейсы для работы с таблицами.

Таблица 3.6 – Функция тестирования добавления записи в таблицу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1 | Добавить запись в таблицу «Типы животных» | 1. В навигации программного средства нажать на пункт меню «Типы животных».   Ввести следующие корректные данные в поля.   1. «Имя типа животного», «Описание типа животного»:   «Млекопитающие», «Животные, которые кормятся молоком».  Нажать на кнопку «Добавить». | 1. Осуществляется переход на пункт   меню «Типы животных», на которой в виде формы находятся поля для заполнения.  «Тип животного» был успешно добавлен в таблицу. | 1. Осуществляется переход на пункт   меню «Типы  животных».   1. Отображается сообщением с успешным добавлением.   Результат представлен в приложении Б на рисунке Б.1. |

Таблица 3.7 – Функция тестирования удаления записи из таблицы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 2 | Удалить запись из таблицы «Типы животных» | 1. В навигации программного средства нажать на пункт меню «Типы животных». 2. Выбрать нужную запись. 3. Нажать на иконку «Удалить». | 1. Осуществляется переход на пункт меню «Типы животных», на которой в виде кнопки находится функция для удаления. 2. «Тип животного» был успешно удалён из таблицы. | 1. Осуществляется переход на пункт меню «Типы животных». 2. Отображается сообщением с успешным добавлением в таблицу.   Результат представлен в приложении Б на рисунке Б.2. |

Таблица 3.8 – Функция тестирования изменения записи в таблице

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 3 | Изменить запись в таблице «Типы животных» | * + 1. В навигации программного средства нажать на пункт меню «Типы животных», далее найти подходящую запись и нажать на иконку «Изменить».   Изменить следующие корректные данные в полях.   1. «Имя типа животного», «Описание типа животного»:   «Птица», «Животные, которые имеют перья».   1. Нажать на кнопку «Изменить» | 1. Осуществляется переход на пункт меню «Типы животных», на которой в виде формы находиться модальное окно для заполнения. 2. «Тип животного» был успешно изменён. | 1. Осуществляется переход на пункт меню «Типы животных». 2. Отображается сообщением с успешным изменением. 3. Результат представлен в приложении Б на рисунке Б.3. |

## Описание справочной системы

В программном средстве разработана справочная система в программе «Dr.Explain».

Иерархия разделов справочной системы представлена на рисунке 3.2.

При выборе необходимого раздела пользователю выдается соответствующая информация.

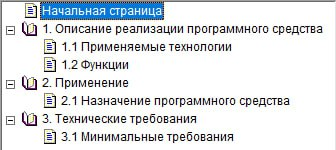


Рисунок 3.2 – Иерархия разделов справочной системы

Например, при выборе «Раздел 2.1» пользователю предоставляется информация о назначении программного средства. Результат представлен на рисунке 3.3.

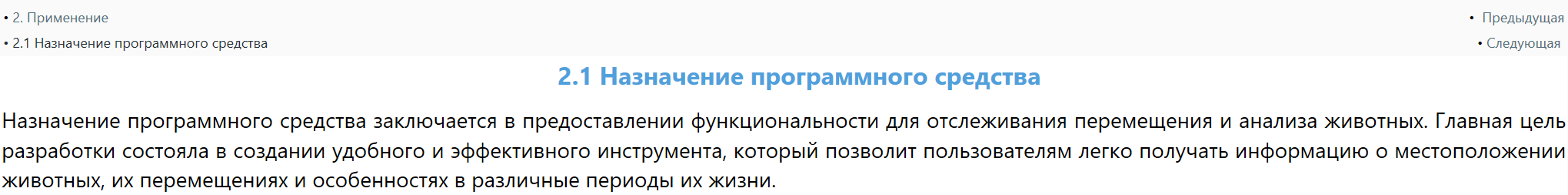


Рисунок 3.3 – Раздел справочной системы «Назначение программного средства»

# Применение

## Назначение программного средства

Назначение программного средства заключается в предоставлении функциональности для анализа месторасположения животных в Национальном парке «Беловежская пуща». Главная цель разработки состояла в создании удобного и эффективного инструмента, который позволит пользователям легко получать информацию о местоположении животных, их перемещениях.

Для достижения этой цели было разработано программное средство, который позволяет пользователям просматривать, а администраторам создавать, изменять и удалять записи о животных, их типах, точках местоположения и местах посещения. Это позволяет администраторам управлять информацией о животных, а пользователями отслеживать их перемещения.

Детали установки будут предоставлены в документации приложения.

Запуск программного средства будет осуществляться через .exe файл.

Область применения разрабатываемого приложения включает зоопарки, ветеринарные клиники, рыбхозы, птицефабрики и другие организации, связанные с уходом за животными и отслеживанием их перемещений. Приложение поможет упростить процессы учета, мониторинга и анализа данных о животных.

Накладываемые ограничения отсутствуют.

## Условия применения

Разрабатываемое приложение на сервере работает под управлением операционной системы Windows 11.

Системные требования для устройства:

* процессор (CPU) c двумя и более физическими ядрами;
* оперативная память (RAM) 2048 Мб;
* свободное место на жестком диске (HDD, SSD или M.2) 1024 Мб.

Для запуска программного средства необходимо установить следующие компоненты:

* платформа .NET Framework 4.7.2;
* система управления базами данных SQL Server 2019.

Для того чтобы развернуть базу данных необходимо присоединить базу данных с помощью SQL Server Management Studio.

Для запуска приложения требуется выгрузить файлы с диска в любую удобную директорию и в файле конфигурации приложения (App.config) изменить строку подключения.

# Заключение

В ходе выполнении курсового проекта было задействовано множество форумов, сайтов, интернет-ресурсов, справочных материалов, методических указаний для разработки программного средства для анализа месторасположения животных в национальном парке «Беловежская пуща».

Для достижения цели курсового проектирования были решены следующие задачи:

* определена вычислительная система, необходимая для создания программного средства;
* разработана физическая и логическая модель данных;
* по модели выполнено проектирование задачи;
* установлены требования и структура программного средства;
* разработана административная и пользовательская часть приложения;
* описаны общие сведения приложения;
* выбрана методика испытаний;
* описан процесс тестирования;
* приведены примеры области применения.

Проект был разработан в среде Visual Studio 2022 на языке C# с использованием Guna.UI2. Использована реляционная база данных Microsoft SQL Server.

Для работы с базой данных выбран СУБД «Microsoft SQL Server Management Studio 19».

Разработка имеет интуитивно понятный графический интерфейс, позволяющий даже с минимальным знанием компьютера использовать данное программного средство.

Программа реализована в полном объеме и в соответствии с заданными требованиями, полностью отлажена и протестирована. Поставленные задачи выполнены.

Среди преимуществ данного программного средства можно выделить следующие аспекты: удобный и понятный интерфейс, легкость использования и отсутствие аналогов в нашем регионе, которые бы могли соперничать с ним по функциональности и удобству.

В процессе разработки данного программного средства были применены и закреплены знания по уже изученному материалу, были отработаны навыки владения методами надежного программирования и эффективности разработки программного обеспечения в Microsoft Visual Studio 2022 с использованием языка программирования C#, разработана база данных средствами системы управления базами данных Microsoft SQL Server 2019.

В планах на будущее предусмотрена разработка мобильной версии данного программного средства. Основная мотивация для этого заключается в обеспечении мобильности пользователей. Мобильная версия приложения предоставляет доступ из любого места, где есть интернет-соединение. Кроме того, она позволяет интеграцию с смартфонами, расширяя возможности использования приложения на различных устройствах.

**Список информационных источников**

1. Михалевич В.Ю. Методические указания для курсового проектирования по учебному предмету КПиЯП / Михалевич В.Ю. – Минск: КБП, 2023
2. Багласова Т.Г. Методические указания по оформлению курсовых проектов, дипломных проектов и отчётов для учащихся специальности 2-40 01 01 «Программное обеспечение информационных технологий» / Т.Г.Багласова, К.О.Якимович. – Минск: КБП, 2023
3. Джеффри Р.Т. CLR VIA C# / Р.Т. Джеффри. – Санкт-Петербург: Питер, 2017
4. Михаил Е. Ф. Библия C# / М. Ф. Евгеньевич. – Санкт-Петербург: Питер, 2019
5. Роберт М.А. Чистый код. Создание, анализ и рефакторинг / М.А. Роберт. – Санкт-Петербург: Питер, 2018
6. Entity Framework [Электронный ресурс] Документация по Entity Framework, 2022 – Режим доступа: https://docs.microsoft.com/en-us/ef/. – Дата доступа 05.01.2022.

1. Guna.UI2 [Электронный ресурс] Руководство Guna.UI. 2024. – Режим доступа: https://gunaui.com/demos/. – Дата доступа 16.03.2024.
2. Visual Studio [Электронный ресурс]. Visual Studio – Быстрая и мощная кросс-платформенная IDE для .NET, 2021. – Режим доступа: https://visualstudio.microsoft.com/ru/.– Дата доступа 10.11.2021.
3. MS SQL SERVER [Электронный ресурс]. MS SQL SERVER – система управления реляционными базами данных, 2021. – Режим доступа: https://www.microsoft.com/ru-ru/sql-server/. – Дата доступа 12.11.2021.
4. .NET Framework [Электронный ресурс] Документация, 2024 – Режим доступа: https://learn.microsoft.com/en-us/dotnet/framework/. – Дата доступа 16.03.2024.

Приложение А

**(обязательное)**

**Текст программных модулей**

Файл ....cs

using

Приложение Б

**(справочное)**

**Результаты работы программы**

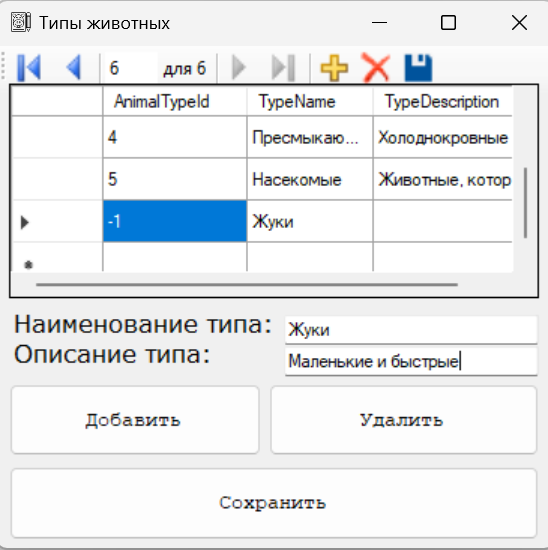


Рисунок Б.1 – Добавление записи в таблицу «Типы животных»

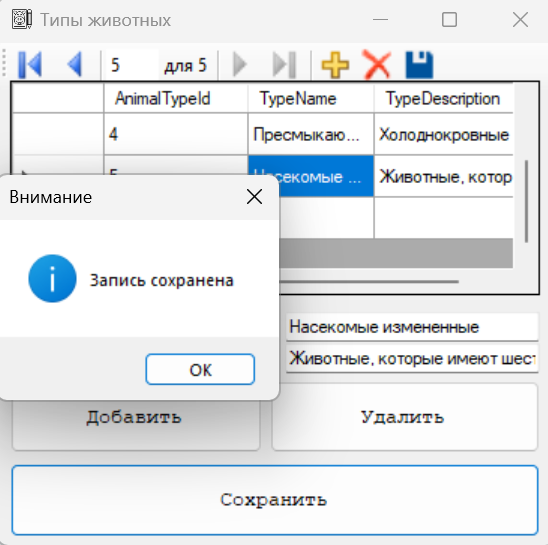


Рисунок Б.2 – Изменение записи в таблице «Типы животных»

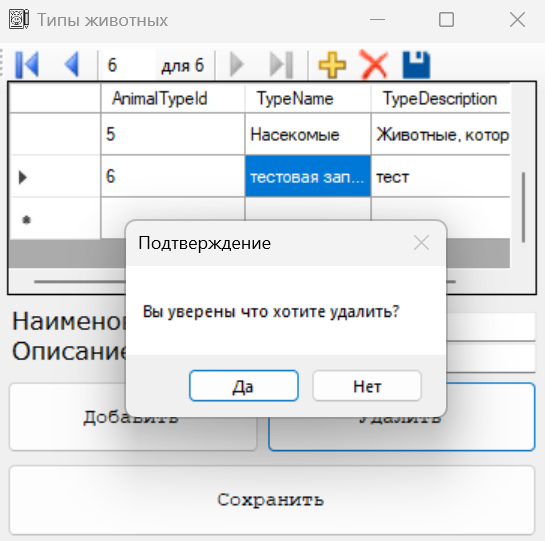


Рисунок Б.3 – Удаление записи из таблицы «Типы животных»

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Ржеутская Н.В.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 1*

*Листов 4*

*Инв.№подл.*

*Подп. и дата*

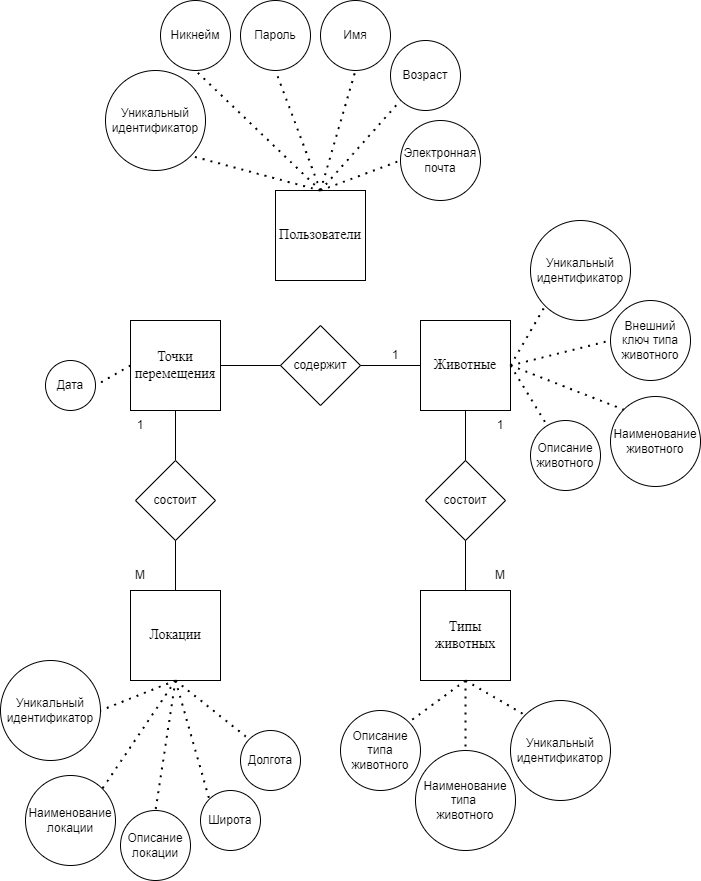
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*КП Т.091004.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КП Т.091004.401 ГЧ*



*Разработка программного средства для анализа месторасположения животных в национальном парке «Беловежская пуща»*

*Диаграмма сущность-связь (концептуальная модель)*

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Ржеутская Н.В.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 2*

*Листов 4*

*Инв.№подл.*

*Подп. и дата*

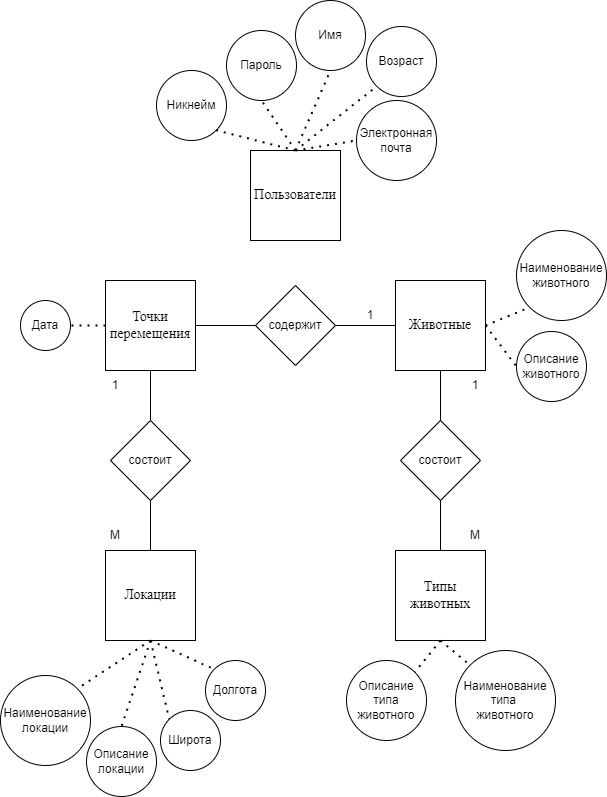
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*КП Т.091004.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КП Т.091004.401 ГЧ*

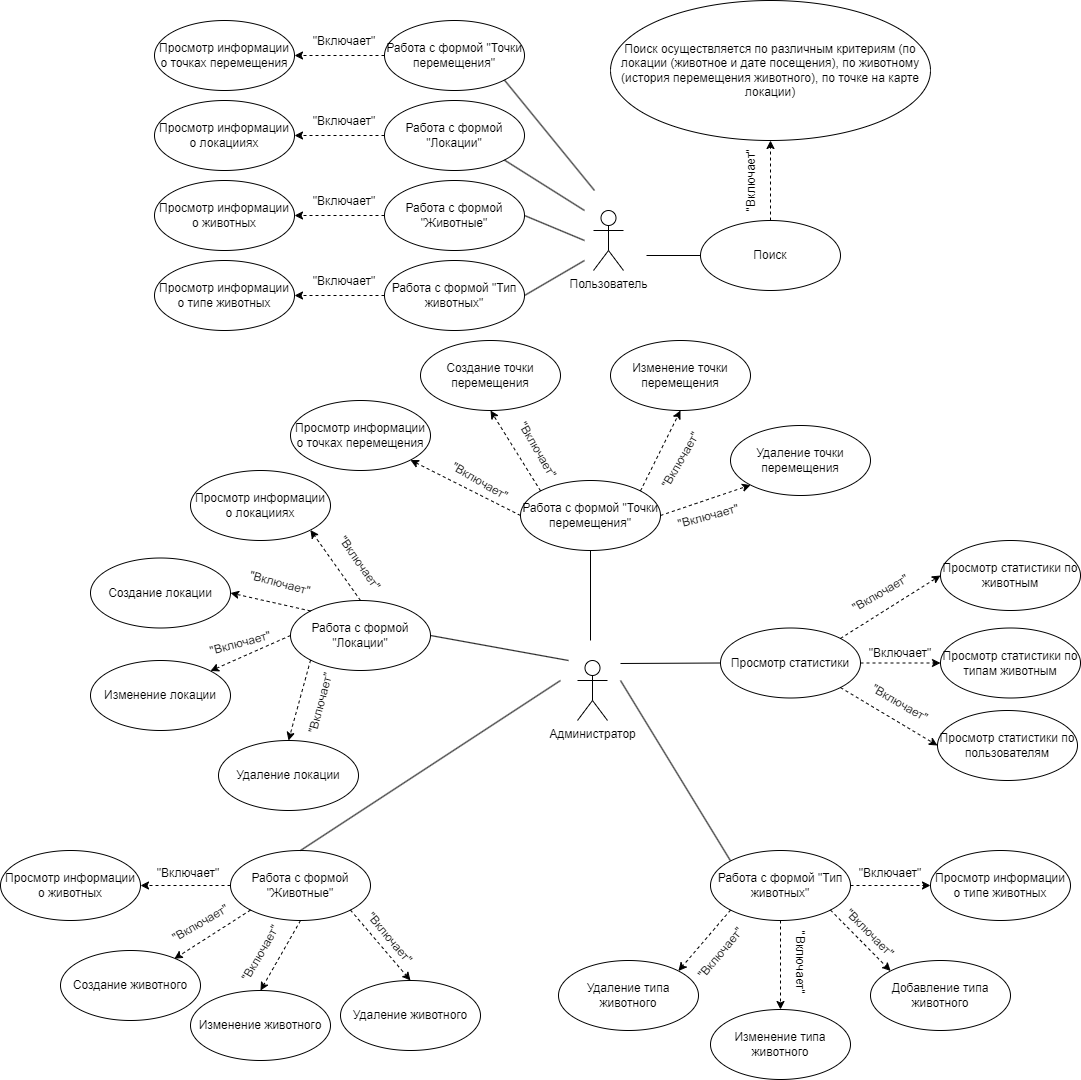


*Веб-приложение «Restful API сервис для отслеживания перемещения животных»*

*Диаграмма классов*

*Разработка программного средства для анализа месторасположения животных в национальном парке «Беловежская пуща»*

*Диаграмма сущность-связь (логическая модель)*



*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Ржеутская Н.В.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 3*

*Листов 4*

*Инв.№подл.*

*Подп. и дата*

*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*КП Т.091004.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КП Т.091004.401 ГЧ*

*Разработка программного средства для анализа месторасположения животных в национальном парке «Беловежская пуща»*

*Диаграмма вариантов*

*использования*

*КБП*

*У*

*Т. Контр.*

*Разраб. Разраб.*

*Зеневич А.О.*

*Провер. Провер.*

*Ржеутская Н.В.*

*Реценз.*

*Утверд.*

*Н. Контр.*

*№ докум.*

*Подпись*

*Дата*

*Изм.*

*Лист*

*Масса*

*Лит.*

*Масштаб*

*Лист 4*

*Листов 4*

*Инв.№подл.*

*Подп. и дата*

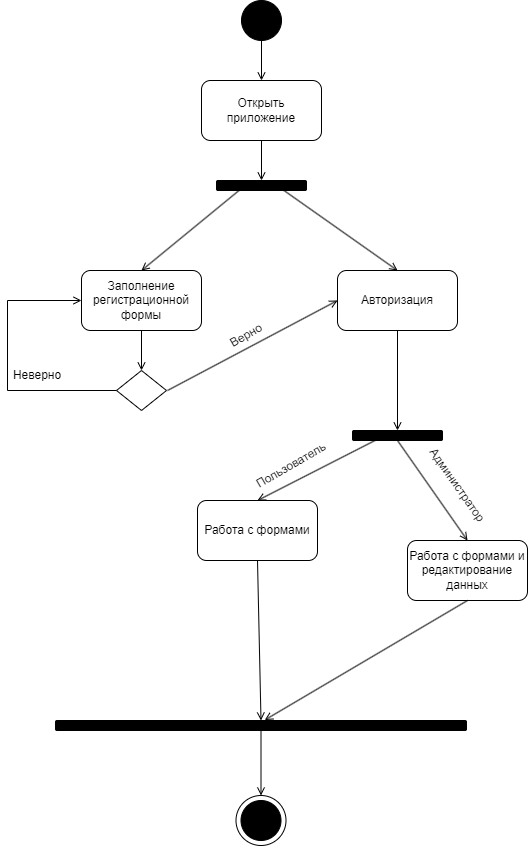
*Взам.инв.№*

*Инв.№дубл.*

*Подп. и дата*

*КП Т.091004.401 ГЧ Т.ХХХХХХ.401 ГЧ*

*КП Т.091004.401 ГЧ*



*Разработка программного средства для анализа месторасположения животных в национальном парке «Беловежская пуща»*

*Диаграмма деятельности*

**Удостоверяющий лист**

электронного документа – курсовой проект

Тема Разработка программного средства для анализа месторасположения животных в Национальном парке «Беловежская пуща»

Обозначение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_КП Т.091004.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Разработчик \_\_\_Зеневич А.О.\_\_\_\_ Руководитель \_\_\_\_Ржеутская Н.В.\_\_\_\_\_\_\_\_

(Ф.И.О.) (Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав электронного документа | Разработчик | Руководитель |
| Пояснительная записка (на бумажном носителе формата А4), ПЗ.docх |  |  |
| ГЧ.docx |  |  |
| Папка с проектом «Animal.Desktop» |  |  |
| Установочный пакет программного средства «Animal.Desktop.exe» |  |  |
| Файлы базы данных «Animals.mdf», «Animals.ldf» |  |  |
| Тип носителя: диск |  |  |

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема «Разработка программного средства для анализа месторасположения животных в Национальном парке «Беловежская пуща» »

КП Т.091004.401

Разработан Зеневич А.О.

Утвержден

Руководитель: Ржеутская Н.В.

Технические средства: AMD Ryzen 7 4.7 ГГц, 16 Гб, SSD 1024 Гб, RXT 3060 8Гб

Программные средства: MS Visual Studio, MS SQL Server Management Studio

**Состав документа:**

Пояснительная записка – ПЗ.docх

Графическая часть – ГЧ.docx

Папка с проектом – Animal.Desktop

Установочный пакет программного средства – Animal.Desktop.exe

Файлы базы данных – Animals.mdf, Animals.ldf

Сведения о защите информации: логин , пароль