Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное  
образовательное учреждение

«Заволжский автомоторный техникум»

(ГБПОУ ЗАМТ)

Специальность: 09.02.03 «Программирование в компьютерных системах»

**Выпускная квалификационная работа**

Студента IV курса группы ПС-19Б

**Дикарева Романа Олеговича**

**Тема:**Разработка автоматизированной информационной системы по учету личностных достижений обучающихся на базе платформы .NET.

Число листов ВКР (прописью) – \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Число страниц пояснительной записки –\_\_\_\_\_\_ страниц

|  |  |
| --- | --- |
| **Допустить к защите:**  Зам. директора по УР  Мокрышева Л.Г.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | **Руководитель**  выпускной квалификационной работы:  Лапина О.И.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  **Рецензент:**  Хмелева Е.О.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Работа защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Заволжье

2023г.

Министерство образования и науки Нижегородской области

Государственное бюджетное профессиональное  
образовательное учреждение

«Заволжский автомоторный техникум»

(ГБПОУ ЗАМТ)

Специальность: 09.02.03 «Программирование в компьютерных системах»

Утверждаю

Зам. директора по УР

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мокрышева Л.Г.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 год

**задание**

**на выпускную квалификационную работу**

Студента IV курса группы ПС-19Б

**Дикарева Романа Олеговича**

**Тема выпускной квалификационной работы:**

Разработка автоматизированной информационной системы по учету личностных достижений обучающихся на базе платформы .NET.

Выпускная квалификационная работа на указанную тему выполняется в следующем объеме:

### Пояснительная записка

Введение

1 Постановка задачи

2 Проектирование

2.1 Обоснование выбора среды разработки

2.2 Выбор способа организации входных и выходных данных

2.3 Проектирование моделей данных

2.4 Алгоритм представления задачи

2.4.1 Детальное проектирование программного обеспечения (прототип)

2.4.2 Описание основных программных модулей

2.5 Применяемые меры защиты информации

3 Тестирование

4 Экономическое обоснование

4.1 Расчет затрат на разработку программы

4.2 Стратегия продвижения программного продукта

4.3 Расчет цены разработанной программы

4.4 Технико-экономические показатели разработки программы

Заключение

Литература

Приложения

#### Дата выдачи задания «06» 2023 года

#### Дата окончания проекта «15» 2023 года

Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Лапина О.И./

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Карпушева И.А./

Содержание

Введение 4

1 Постановка задачи 6

2 Проектирование 7

2.1 Обоснование выбора среды разработки 7

2.2 Выбор способа организации входных и выходных данных 9

2.3 Проектирование моделей данных 17

2.4 Алгоритм представления задачи 22

2.4.1 Детальное проектирование программного обеспечения (прототип) 19

2.4.2 Описание основных программных модулей 34

2.5 Применяемые меры защиты информации 40

3 Тестирование 45

4 Экономическое обоснование 52

4.1 Расчет затрат на разработку программы 54

4.2 Стратегия продвижения программного продукта 56

4.3 Расчет цены разработанной программы 57

4.4 Технико-экономические показатели разработки программы 58

Заключение 59

Литература 61

Приложения

**Введение**

Огромную роль в формировании будущего специалиста играет не только учебная, но и воспитательная работа. Ее основная цель - воспитание гармоничной личности, сочетающей в себе владение профессиональными компетенциями с высоким уровнем нравственного сознания, подготовка востребованного специалиста к жизни в социуме. Для реализации разрабатываются программы и планы с различными мероприятиями для формирования личности. При этом классный руководитель должен вести учет количества участвующих студентов в мероприятиях, их результаты и ежегодно представлять отчет по личностям результатам обучающихся в учебную часть.

В настоящее время учет ведется в бумажном виде. Это сбор ксерокопий грамот и дипломов, обучающихся и составление бумажного портфолио. Иногда используется частичная автоматизация с помощью MS Excel. Но она требует определенных навыков работы в MS Excel, которые отсутствуют у большей части классных руководителей. Для учета нужна автоматизированная информационная система.

Целью разработки является создание автоматизированной информационной системы по учету личностных достижений обучающихся на базе платформы .NET.

В процессе проектирования необходимо решить следующие задачи:

* сформулировать требования к программе;
* построить структуру базы данных;
* выбрать среду разработки;
* реализовать программный продукт;
* протестировать программный продукт;
* рассчитать технико-экономические показатели;
* разработать техническую и эксплуатационную документацию.

Личностные достижения студентов являются важным аспектом их учебной жизни. Они отражают прогресс и успехи учеников в процессе обучения, а также способствуют развитию их личности. Для фиксации этих достижений существует автоматизированная информационная система учета, которая позволяет собирать всю необходимую информацию по личным достижениям студентов в одном месте. Это экономит время на составление портфолио на каждого студента учебного заведения, повышает точность учета ведения личностных и групповых достижений.

Система делает возможным создание портфолио на всю группу или на конкретного студента, что позволяет продемонстрировать основные положительные качества и достижения. Документация из портфолио студента может быть использована последующими учебными учреждениями и/или будущими работодателями.

**1 Постановка задачи**

Разработать автоматизированную информационную систему по учету личностных достижений обучающихся, на базе платформы .NET. Для разработки приложения использовать язык программирования высокого уровня. Для хранения информации использовать базу данных по усмотрению разработчика.

База должна содержать информацию: о группе, студентах группы, пользователях АИС, о мероприятиях, конкурсах, о личностных результатах, участиях и наградах.

С АИС должны работать три пользователя: системный администратор, классный руководитель и студент (староста).

*Студент* должен иметь возможность решать следующие задачи:

* + заполнять данные о группе, курсах обучения, студентах группы и мероприятия.

*Классный руководитель* АИС имеет возможность:

* + заполнять данные о группе, курсах обучения, студентах группы и мероприятия;
  + просматривать формы: учета мероприятий и мониторинг группы, используя функции поиска, сортировки и фильтрации;
  + формировать отчеты «Учет мероприятий» и «Мониторинг группы» для экспорта в приложения MS Excel 2007.

*Системный администратор* АИС имеет возможность:

* разграничивать права доступа к разделам базы;
* вести структуру базы данных;
* заполнять справочники БД с помощью импорта данных;
* выполнять резервное копирование и восстановление базы данных.

Количество форм информационной системы определяет разработчик. В АИС предусмотреть выполнение всех функций, указанных в общих требованиях к программному продукту.

**2 Проектирование**

2.1 Обоснование выбора среды разработки

Для создания данного продукта можно использовать следующие среды разработки:Microsoft Visual Studio Enterprise, [Visual Studio Code](https://code.visualstudio.com/) и[JetBrains](https://www.eclipse.org/) Rider.

*Microsoft Visual Studio Enterprise* представляет собой комплекс программных продуктов от Microsoft, который включает в себя интегрированную среду разработки ПО. Пользователи могут использовать эти продукты для создания как консольных приложений, так и приложений с графическим интерфейсом, включая поддержку технологии Windows Forms, а также веб-сайтов, веб-приложений и веб-служб. Этот инструмент имеет множество функций и доступны качественные плагины, которые могут расширить возможности приложения и подключить другие языки. Существует бесплатная версия этого ПО. Для работы в последних версиях Visual Studio могут потребоваться дополнительные ресурсы компьютера, что может повлиять на производительность системы, а также на работу с программой в целом.

*Visual Studio Code* – это мощный и бесплатный инструмент для разработчиков, который предоставляет полноценный набор функций и легко расширяем. Он отлично подходит для создания не корпоративных приложений и не требует больших ресурсов компьютера. Некоторые из доступных дополнений, однако, могут иметь низкое качество сборки и не всегда работать должным образом. Тем не менее, благодаря широкому спектру возможностей и простоте использования, Visual Studio Code остается одним из самых популярных инструментов среди разработчиков.

*JetBrains Rider –* это кроссплатформенная среда разработки (IDE) для создания приложений на .NET Framework, .NET Core и Mono. Она разработана компанией JetBrains, которая также известна своими другими продуктами, такими как IntelliJ IDEA, PyCharm и ReSharper.

Rider является полнофункциональной IDE, которая позволяет разрабатывать приложения на C#, VB.NET, F# и других языках программирования, работающих на платформе .NET. Она включает в себя редактор кода с подсветкой синтаксиса, автодополнением и функцией рефакторинга кода, отладчик, инструменты для управления проектами и сборкой приложений, а также интеграцию с системами контроля версий, такими как Git и Subversion.

Одним из основных преимуществ Rider является его интуитивно понятный интерфейс. Он легко конфигурируется и настраивается под нужды пользователя. Благодаря этому, процесс разработки может быть ускорен и упрощен в несколько раз. Кроме того, Rider имеет мощный набор функций для отладки, автоматического завершения кода, рефакторинга кода и других операций, что также способствует повышению скорости и качества разработки.

Исходя из вышеперечисленных инструментальных средств, была выбрана среда разработки JetBrainsRider, так как данная среда разработки имеет достаточный функционал для написания данного проекта.

JetBrainsRiderпозволяет программировать на следующих языках программирования: C#, VB.NET и F#.

Для разработки автоматизированной информационной системы идеально могут подойти следующие языки программирования:

C# – это объектно-ориентированный язык программирования, который также поддерживает компонентно-ориентированное программирование. Современные приложения все больше используют программные компоненты в виде автономных пакетов, которые предоставляют отдельные функциональные возможности. Основным преимуществом таких компонентов является их модель программирования, которая включает в себя свойства, методы и события. Кроме того, они имеют атрибуты, которые предоставляют декларативную информацию о компоненте, включая его документацию. C# обладает языковыми конструкциями, которые непосредственно поддерживают эту концепцию работы, что делает его идеальным для создания и использования программных компонентов.

*Python –* это высокоуровневый, интерпретируемый язык программирования, который создан с целью быть простым в использовании, понятным и эффективным. Одной из главных особенностей Python является его читабельный и лаконичный синтаксис, который облегчает написание кода и повышает его понятность для других программистов. Кроме того, он поддерживает множество парадигм программирования, включая объектно-ориентированное, функциональное и процедурное программирование.

Один из минусов языка Python заключается в его относительно низкой производительности по сравнению с компилируемыми языками программирования, такими как C# или Java.

*Java* – одной из основных принципов языка Java является "написание один раз, запуск везде", что означает, что код, написанный на Java, может быть запущен практически на любой операционной системе (Windows, macOS, Linux и т.д.), где установлена виртуальная машина Java. Одним из минусов языка Java является его относительно медленная скорость выполнения кода по сравнению с некоторыми другими языками. Также, использование виртуальной машины Java может потребовать дополнительных ресурсов компьютера, что может повлиять на производительность системы и приложения.

Исходя из выше перечисленных языков программирования для разработки был выбран язык С#, так как это объектно-ориентированный высокоуровневый язык программирования, то есть он подходит для написания графических приложений. Работа с внутренностями .Net не требуется, сроки решения задачи не допускают этого, гораздо больше внимания следует уделить непосредственно задаче, а не техническим деталям. Исходя из этого, предпочтение было отдано в сторону С#, так как использование языка Java для выполнения выпускной квалификационной работы нет необходимости, а Python не соответствует постановке задачи.

2.2 Выбор способа организации входных и выходных данных

Для организации входных и выходных данных необходима база данных. JetBrainsRider позволяет использовать следующие базы данных: MS Access,MуSQL,MongoDBи другие.

*MSAccess*– одна из самых популярных систем управления базами данных в мире. Она предназначена для использования на персональных компьютерах и обладает широкими возможностями, которые делают ее удобной и функциональной для различных задач.

Один из минусов MS Access заключается в том, что у нее может быть относительно слабая защита данных. Пользователи могут иметь доступ к базе данных без авторизации или с использованием простого пароля, что может привести к несанкционированному доступу к конфиденциальной информации. Более того, у MS Access может быть недостаточно функций и инструментов для обеспечения должного уровня безопасности, таких как шифрование данных и контроль доступа на уровне записей. Это может стать серьезной проблемой для компаний и организаций, хранящих большие объемы информации.

*MySQL –*это мощная, свободная реляционная система управления базами данных, которая обладает широкими возможностями и высокой производительностью. Разработчиком MySQL является корпорация Oracle, которая усердно работает над постоянным улучшением и развитием продукта.

Кроме того, MySQL позволяет создавать масштабируемые и высокопроизводительные решения, которые способны обрабатывать большие объемы данных. Он поддерживает множество функций, таких как индексирование, триггеры и хранимые процедуры, которые помогают ускорить обработку и анализ данных.

Еще одним преимуществом MySQL является его гибкость. Он может быть развернут на большинстве операционных систем и поддерживает множество языков программирования, включая C#, Java и Python. Это делает его идеальным выбором для разработчиков, которые хотят создавать приложения на различных платформах.

*MongoDB*– это документоориентированная система управления базами данных, которая позволяет хранить и обрабатывать большие объемы данных.

Одним из главных преимуществ MongoDB является ее гибкость и масштабируемость. В отличие от реляционных баз данных, MongoDB не имеет фиксированных схем, что позволяет сохранять данные в произвольном формате.

Главным из минусов MongoDB является относительно низкая производительность при выполнении сложных запросов и операций, таких как объединение таблиц и агрегирование данных. Это может быть особенно заметно при работе с большими объемами данных, что может привести к увеличению времени ответа системы и снижению скорости работы приложения. Кроме того, MongoDB не поддерживает полноценную транзакционность, что может создавать проблемы при обработке транзакций в высоконагруженных средах.

Исходя из выше перечисленных вариантов, принято решение использовать реляционную систему управления базами данных *MySQL*, так как ее возможностей хватает для выполнения поставленной задачи. Помимо этого имеется практический практический навык, позволит в песпективе перейти к клиент-серверной версии.

База данных состоит из 8 таблиц,7 из которых имеют связь в соответствии с рисунком 2.1.

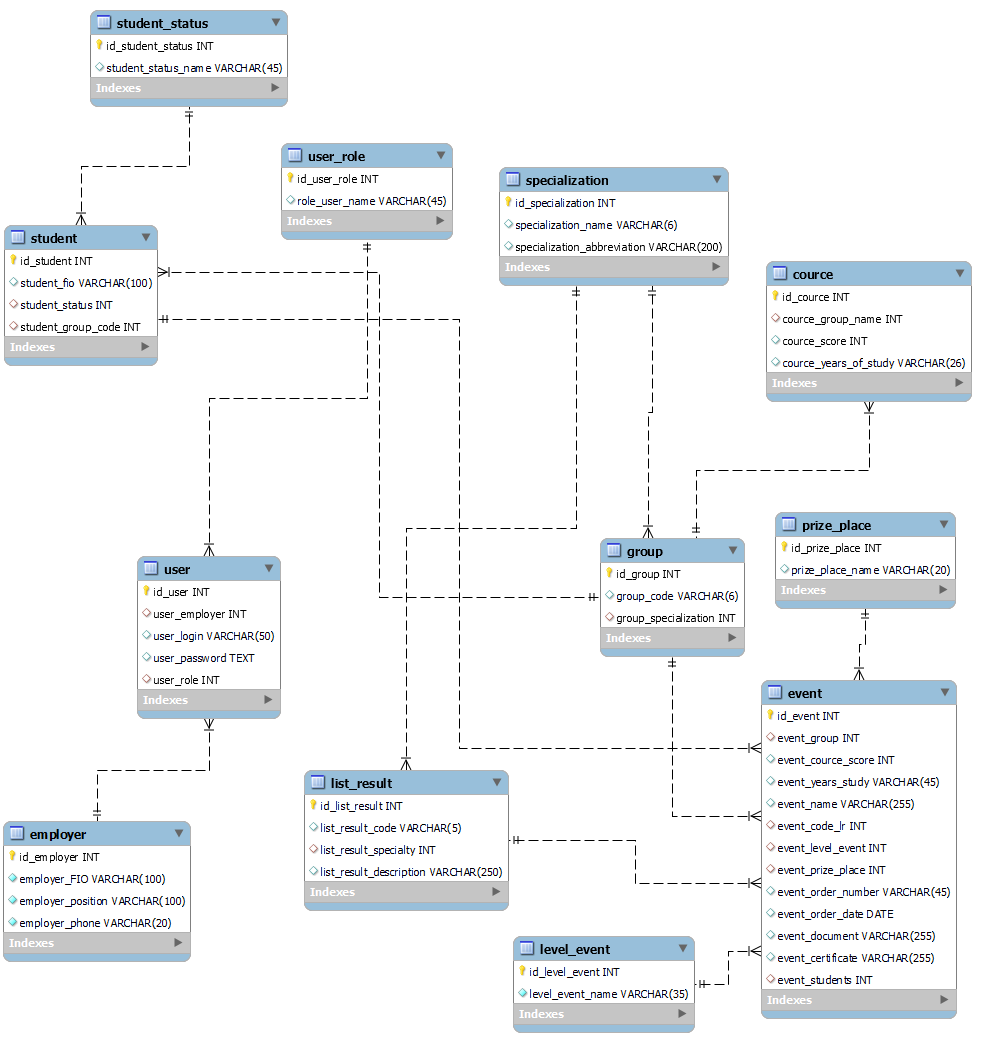


Рисунок 2.1 – Структура базы данных

Структура таблиц базы данных представлена таблицами 2.1 – 2.12

Таблица 2.1 – Структура таблицы «cource» (Курс)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификаторкурса | id\_cource | Счетчик | длинное целое число, заполняется автоматически |
| Наименование группы | courсe\_group\_name | Числовой | ограничение в 100 знаков |
| Номер курса | courсe\_score | Числовой | целое число |
| Годы обучения | cource\_years\_of\_study | Текстовый | ограничение в 45 знаков |

Таблица 2.2 – Структура таблицы «employer» (Сотрудник)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца  (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор сотрудника | id\_employer | Счетчик | длинное целое число, заполняется автоматически |
| ФИО сотрудника | employer\_FIO | Текстовый | ограничение в 150 знаков |
| Должность сотрудника | employer\_position | Числовой | ограничение в 100 знаков |
| Телефон сотрудника | employer\_phone | Текстовый | ограничение в 20 знаков |

Таблица 2.3 – Структура таблицы «event» (Мероприятие)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор мероприятия | id\_event | Счетчик | длинное целое число, заполняется автоматически |
| Идентификатор группы | event\_group | Числовой | целое число, ключ, связанный с таблицей «group» |
| Номер курса | event\_cource\_score | Числовой | целое число |

Продолжение таблицы 2.3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Учебный год | event\_years\_study | Числовой | ограничение в 13 знаков |
| Идентификатор кода ЛР | event\_code\_lr | Числовой | целое число, ключ, связанный с таблицей «list\_result» |
| Идентификатор уровня мероприятия | event\_level\_event | Числовой | целое число, ключ, связанный с таблицей «level\_event» |
| Идентификатор призового места | event\_prize\_place | Числовой | целое число, ключ, связанный с таблицей «prize\_place» |
| Идентификатор номера приказа | event\_order\_number | Числовой | целое число, ключ, связанный с таблицей «role» |
| Дата приказа | event\_order\_date | Числовой | целое число, ключ, связанный с таблицей «role» |
| Фотография грамоты | event\_document | Текстовый | ограничение в 255 знаков |
| Фотография сертификата | event\_certificate | Текстовый | ограничение в 255 знаков |
| Студенты, принимавшие участие в мероприятии | event\_students | Числовой | целое число, ключ, связанный с таблицей «student» |

Таблица 2.4 – Структура таблицы «group» (Группа)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор группы | id\_group | Счетчик | длинное целое число, заполняется автоматически |
| Наименование группы | group\_code | Текстовый | ограничение в 6 знаков |
| Аббревиатура специализации | group\_specialization | Числовой | целое число, ключ, связанный с таблицей «specialization» |

Таблица 2.5 – Структура таблицы «level\_event» (Уровень мероприятия)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор уровня мероприятия | id\_level\_event | Счетчик | целое число, заполняется автоматически |
| Наименования уровня мероприятия | level\_event\_name | Текстовый | ограничение в 35 знаков |

Таблица 2.6 – Структура таблицы «list\_result» (Перечень ЛР)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор номера ЛР | id\_list\_result | Счетчик | целое число, заполняется автоматически |
| Наименования кодаЛР | list\_result\_code | Текстовый | ограничение в 5 знаков |
| Специализация для ЛР | list\_result\_specialty | Числовой | целое число, ключ, связанный с таблицей «specialization» |
| Описание для ЛР | list\_result\_description | Текстовый | ограничение в 250 знаков |

Таблица 2.7 – Структура таблицы «prize\_place» (Призовое место)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор призового места | id\_prize\_place | Счетчик | целое число, заполняется автоматически |
| Наименования призового места | prize\_place\_name | Текстовый | ограничение в 20 знаков |

Таблица 2.8 – Структура таблицы «specialization» (Специальность)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификаторспециальности | id\_specialization | Счетчик | длинное целое число, заполняется автоматически |

Продолжение таблицы 2.8

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Наименование специальности | specialization\_name | Текстовый | ограничение в 6 знаков |
| Аббревиатура специальности | specialization\_abbreviation |  | ограничение в 200 знаков |

Таблица 2.9 – Структура таблицы «student» (Студент)

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор студента | id\_student | Счетчик | длинное целое число, заполняется автоматически |
| ФИО студента | student\_fio | Текстовый | ограничение в 150 знаков |
| Статус обучения студента | student\_status | Числовой | однозначноецелое число |
| Номер группы | student\_group\_code | Числовой | целое число, ключ, связанный с таблицей «group» |

Таблица 2.10 – Структура таблицы «student\_status» (Статус студента)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| Идентификатор статуса студента | id\_student\_status | Счетчик | целое число, заполняется автоматически |
| Статус обучения студента | student\_status\_name | Текстовый | ограничение в 45 знаков |

Таблица 2.11 – Структура таблицы «user» (Пользователь)

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца (в БД) | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор пользователя | id\_user | Счетчик | целое число, заполняется автоматически |
| ФИО пользователя | user\_employer | Числовой | целое число, ключ, связанный с таблицей «employer» |

Продолжение таблицы 2.11

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Логин пользователя | user\_login | Текстовый | ограничение в 50 знаков |
| Пароль пользователя | user\_password | Текстовый | ограничение в 512 знаков |
| Доступ пользователя | user\_role | Числовой | целое число, ключ, связанный с таблицей «role» |

Таблица 2.12 – Структура таблицы «user\_role» (Пользователь)

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор роли пользователя | id\_user\_role | Текстовый | длинное целое число, заполняется автоматически |
| Наименование роли пользователя | role\_user\_name | Числовой | ограничение в 45 знаков |

2.3 Проектирование моделей данных

При проектировании автоматизированной информационной системы по учету личностных достижений обучающихся были созданы диаграммы вариантов использования и деятельности. Диаграммы вариантов использования позволяют определить функциональное назначение системы и ее возможности при выполнении различных видов работ. Также они помогают определить функциональные требования к ПС, что является важным этапом в процессе ее проектирования и разработки.

Важно отметить, что автоматизированные информационные системы по учету личностных достижений обучающихся имеют большое значение для современного образовательного процесса. Они позволяют более эффективно и точно отслеживать успеваемость студентов, а также сохранять ценную информацию о каждом ученике на протяжении всего обучения. Это в свою очередь помогает учителям и родителям более глубоко понимать потребности каждого ученика и разрабатывать индивидуальные подходы к его обучению. Диаграмма вариантов использования автоматизированной информационной системы по учету личностных достижений обучающихся на рисунке 2.2.

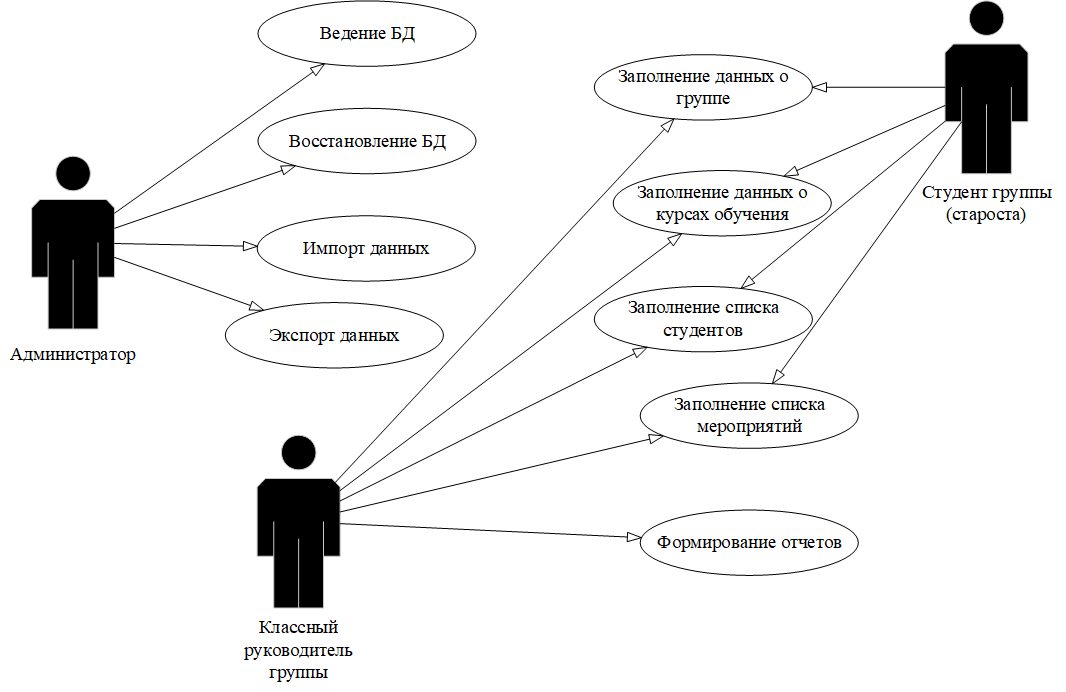


Рисунок 2.2 - Диаграмма вариантов использования

Сценарий использования в программном обеспечении и системном проектировании – это способ использования, случай использования или пользовательский сценарий. Он определяет поведение системы в ответ на внешние запросы и описывает, какие действия могут быть выполнены с данной системой. С помощью методики сценариев использования можно выявить функциональные требования к системе. Сценарий диаграммы представлен в таблице 2.9.

Таблица 2.9 Сценарий вариантов использования

|  |  |
| --- | --- |
| Действие актеров | Отклик |
| 1 | 2 |
| Варианты использования | Ведение БД, заполнение информации о группе и мероприятиях, формированиеотчета |
| Актеры | Администратор, студент, классный руководитель |

Продолжение таблицы 2.9

|  |  |
| --- | --- |
| 1 | 2 |
| Краткое описание | Администратор отвечает за ведение базы данных, обеспечивая ее стабильность и безопасность. Он может проводить экспорт данных из БД, а также восстанавливать данные в случае каких-либо сбоев.  Студент использует АИС для заполнения информации о группе, заносить информацию о студентах и заполнять список мероприятий.  Классный руководитель использует АИС для тех же функций, что и студент. Заполняет информацию о группе, заносит информацию о студентах и заполняет список мероприятий. Однако, помимо этого имеет доступ к сформированным отчетам по учету мероприятий и мониторингу группы. |
| Цель | Занесение информации для формирования отчета к печати. |
| Тип | Базовый |
| Ссылки на другие варианты использования | - экспорт сформированных отчетов;  - создание и изменение списка мероприятий |

Актер - это существо или объект, находящийся внутри или за пределами системы и влияющий на нее либо подверженный ее воздействию. Актер может быть человеком, устройством, другой системой или временем. Человек в реальном мире может играть несколько ролей и выполнять различные действия в отношении системы.

Использование сценариев в процессе разработки зависит от методологии, используемой при разработке. Некоторые методологии требуют только краткого обзора сценария, если не требуется актера для достижения цели. Цель описывает то, что пользователь намеревается достигнуть с помощью данного сценария, в соответствии с таблицей 2.10. Главная цель сценария - решить поставленные задачи.

Таблица 2.10Последовательность действий актера (администратор)

|  |  |
| --- | --- |
| Действия актеров | Отклик |
| 1. Администратор запускает программу | 1. Открытие формы авторизации |
| 1. Администратор вводит логин и пароль | 1. Открытие главного меню |
| 1. Администратор переходит на окно «Справочники» | 1. Заполняет справочники |
| 1. Администратор возвращается в главное меню и переходит на окно «Пользователи | 1. Открытие окна «Пользователи» |
| 1. Администратор добавляет пользователей | 1. Выбор сотрудника, ввод логина и пароля |
| 1. Администратор возвращается в главное меню и переходит на окно «Специальные возможности | 1. Производит экспорт БД |

Таблица 2.11 Последовательность действий актеров (студент и классный руководитель)

|  |  |
| --- | --- |
| Действия актеров | Отклик |
| 1. Актер запускает программу | 1. Открытие окна авторизации |
| 1. Актер вводит логин и пароль | 1. Открытие главного меню |
| 1. Актер переходит на окно «Группы» и заполняет информацию о группе | 1. Открытие окна «Группы» и заполнение информации |
| 1. Актер переходит на окно «Список студентов» и заполняет информацию о студентах | 1. Окрытие окна «Список студентов» и заполнение информации |
| 1. Актер переходит на окно «Курсы» и заполняет информацию о курсах обучения группы | 1. Окрытие окна «Курсы» и заполнение информации |
| 1. Актер переходит на окно «Мероприятия» и заполняет список мероприятий с участниками | 1. Открытие окна «Мероприятия» и заполнение информации |
| 1. Классный руководитель переходит на окно «Отчеты» и сформировывает отчеты для печати | 1. Экспорт отчетов по учету мероприятий и мониторинга по группе в MS Excel |

Диаграмма деятельности - это одна из диаграмм UML, которая показывает расчленение определенной деятельности на ее составные части. Деятельность - это спецификация исполнительного поведения, которое может быть выполнено последовательно или параллельно с помощью вложенных видов деятельности и отдельных действий, связанных между собой потоками, которые идут от выходов одного узла к входам другого.

Диаграмма деятельности широко используется для моделирования бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. Программная диаграмма деятельности представленана рисунке 2.3.

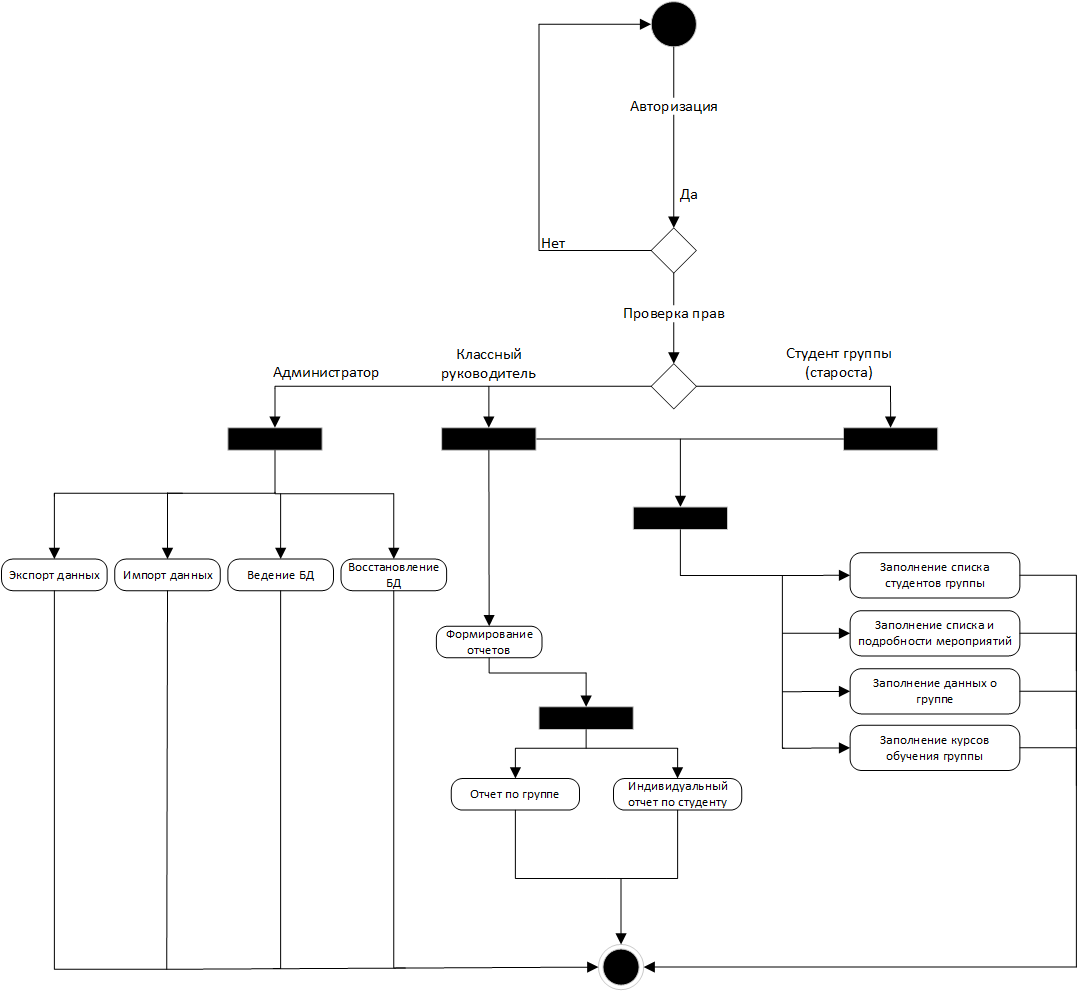


Рисунок 2.3 - Диаграмма деятельности

2.4 Алгоритм представления задачи

2.4.1 Детальное проектирование пользовательского интерфейса

Детальное проектирование ПО – это подробное описание компонентов программного обеспечения и интерфейсов между ними.

Прототипирование программного обеспечения – этап разработки программного обеспечения, процесс создания прототипа программы – макета(черновой, пробной версии) программы, обычно – с целью проверки пригодности предлагаемых для применения концепций, архитектурных или технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

На рисунке 2.4 показаны формы аторизации

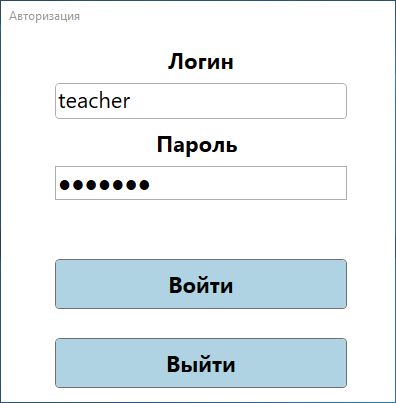
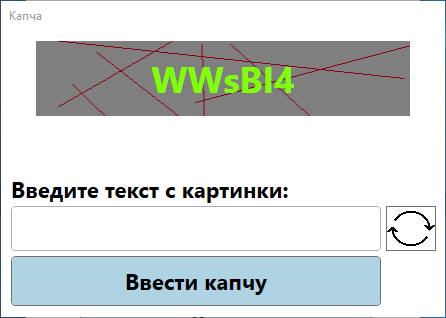


Рисунок 2.4 и 2.5 – Формы авторизации

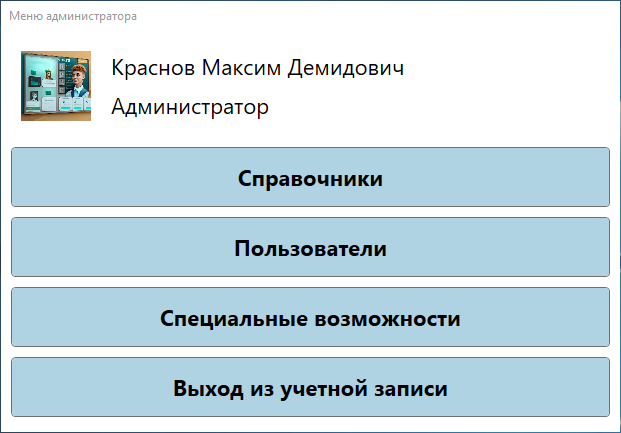


Рисунок Б.2.6 – Главное меню администратора



Рисунок Б.2.7 – Главное меню студента группы (старосты)



Рисунок Б.2.8 – Главное меню администратора

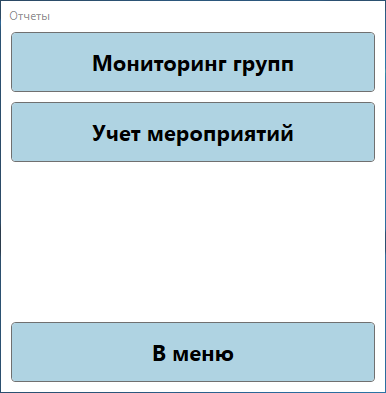


Рисунок Б.2.9 – Окно «Отчеты» в режиме классного руководителя

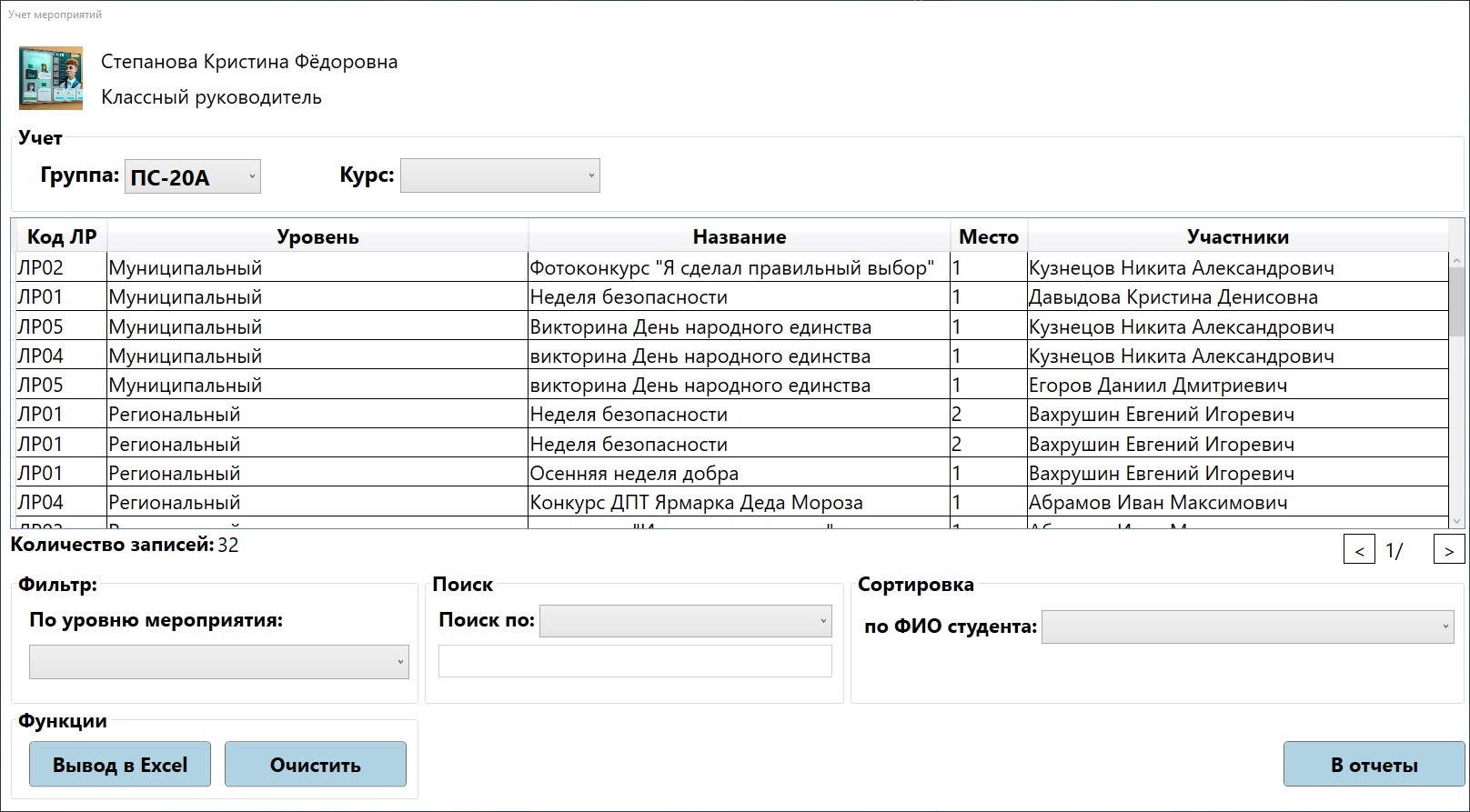


Рисунок Б.2.10 – Окно «Учет мероприятий»

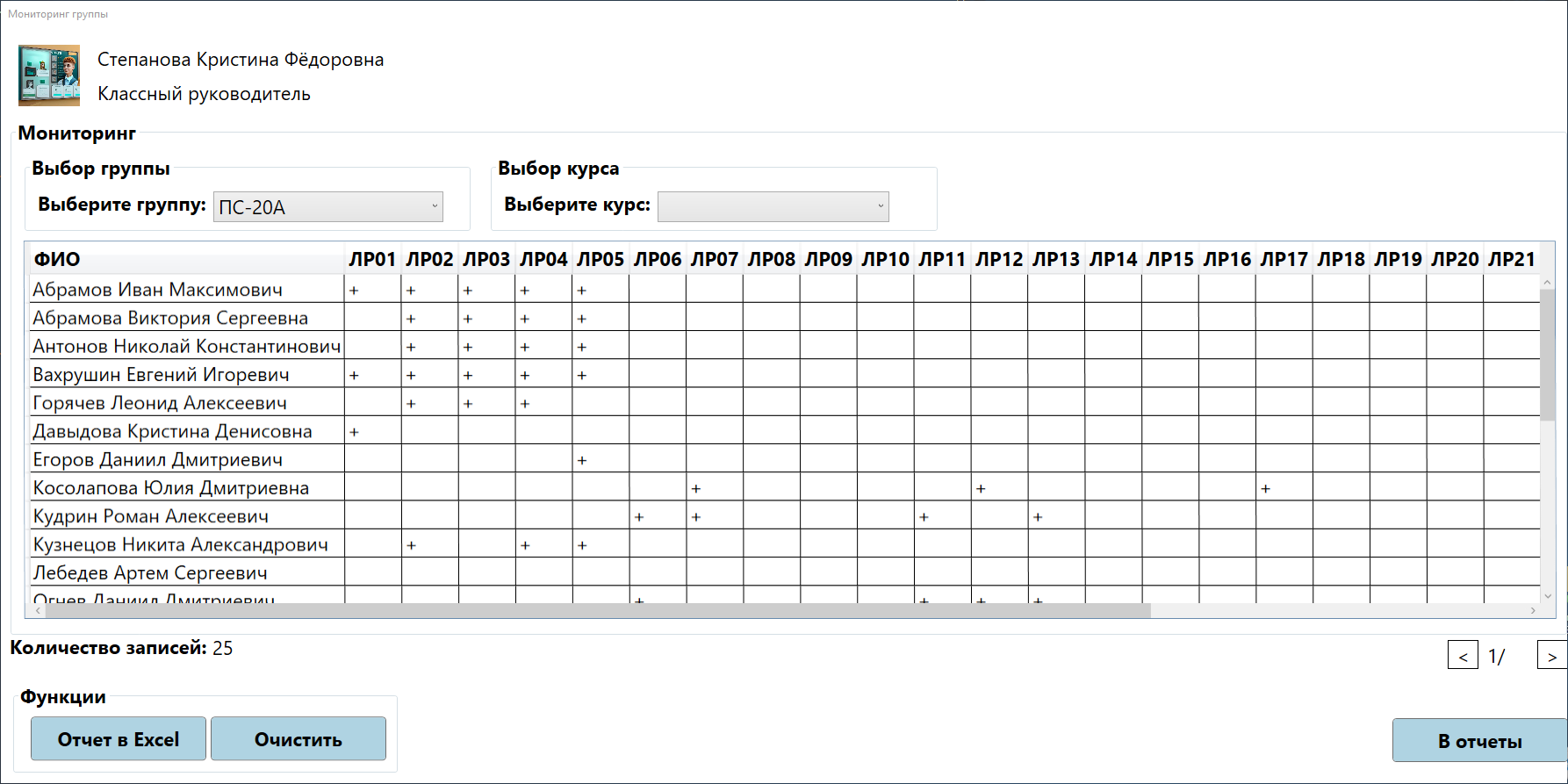


Рисунок Б.2.11 – Окно «Мониторинг группы»

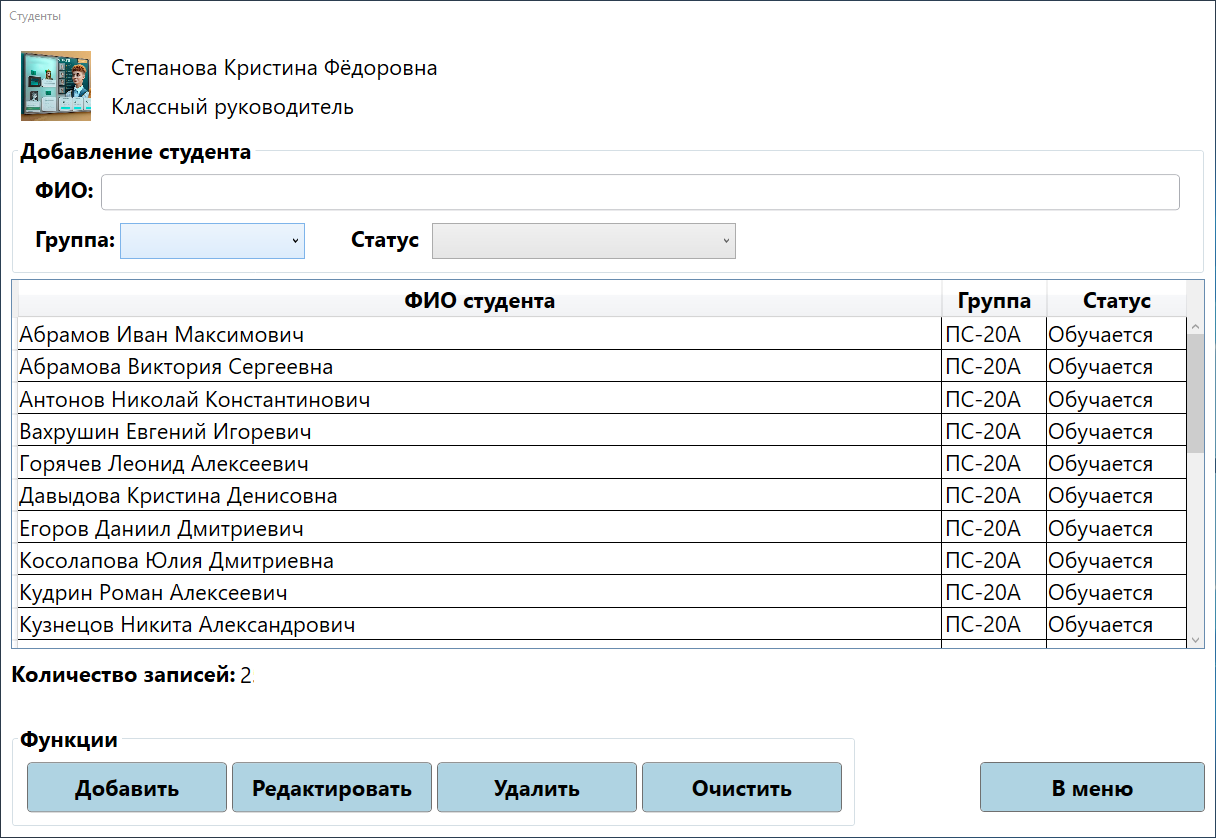


Рисунок Б.2.12 – Окно «Студенты»

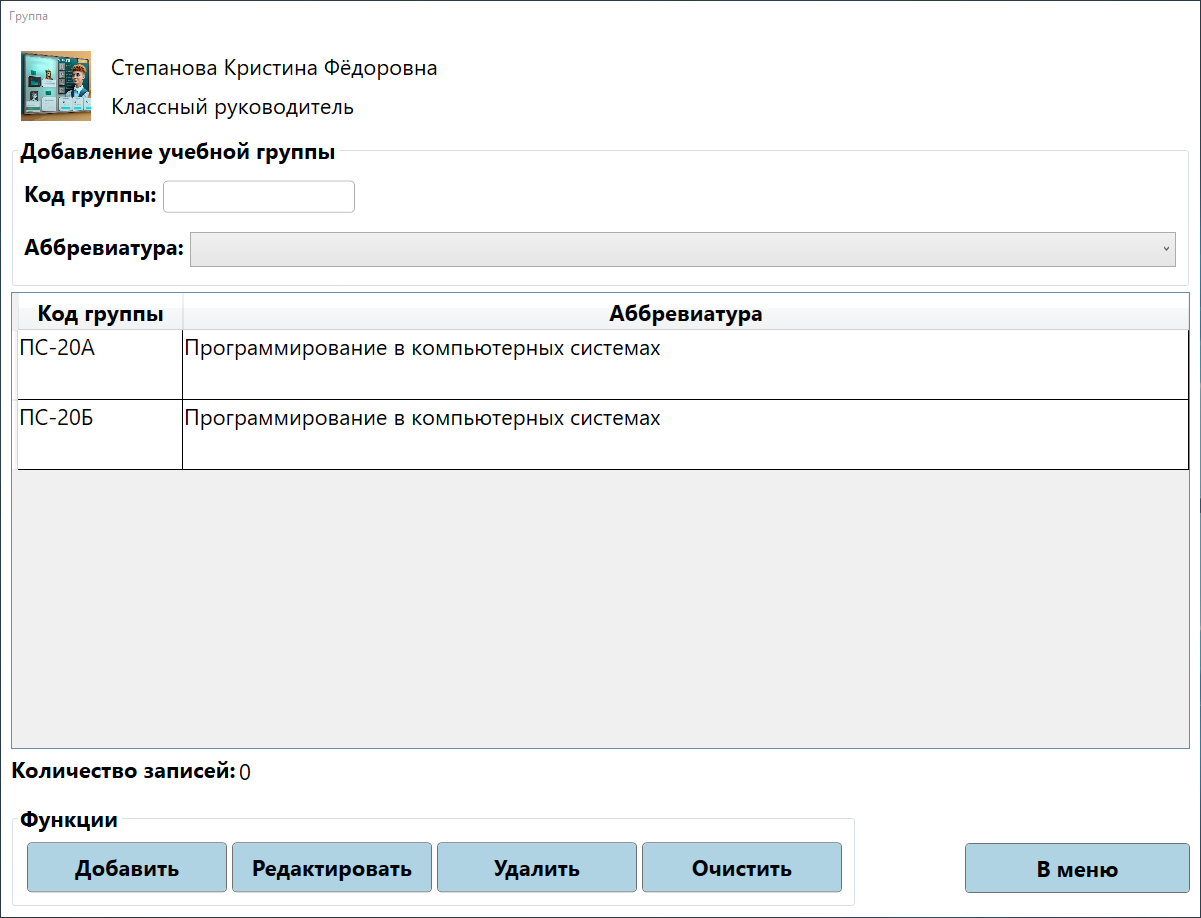


Рисунок Б.2.13 – Окно «Группа»

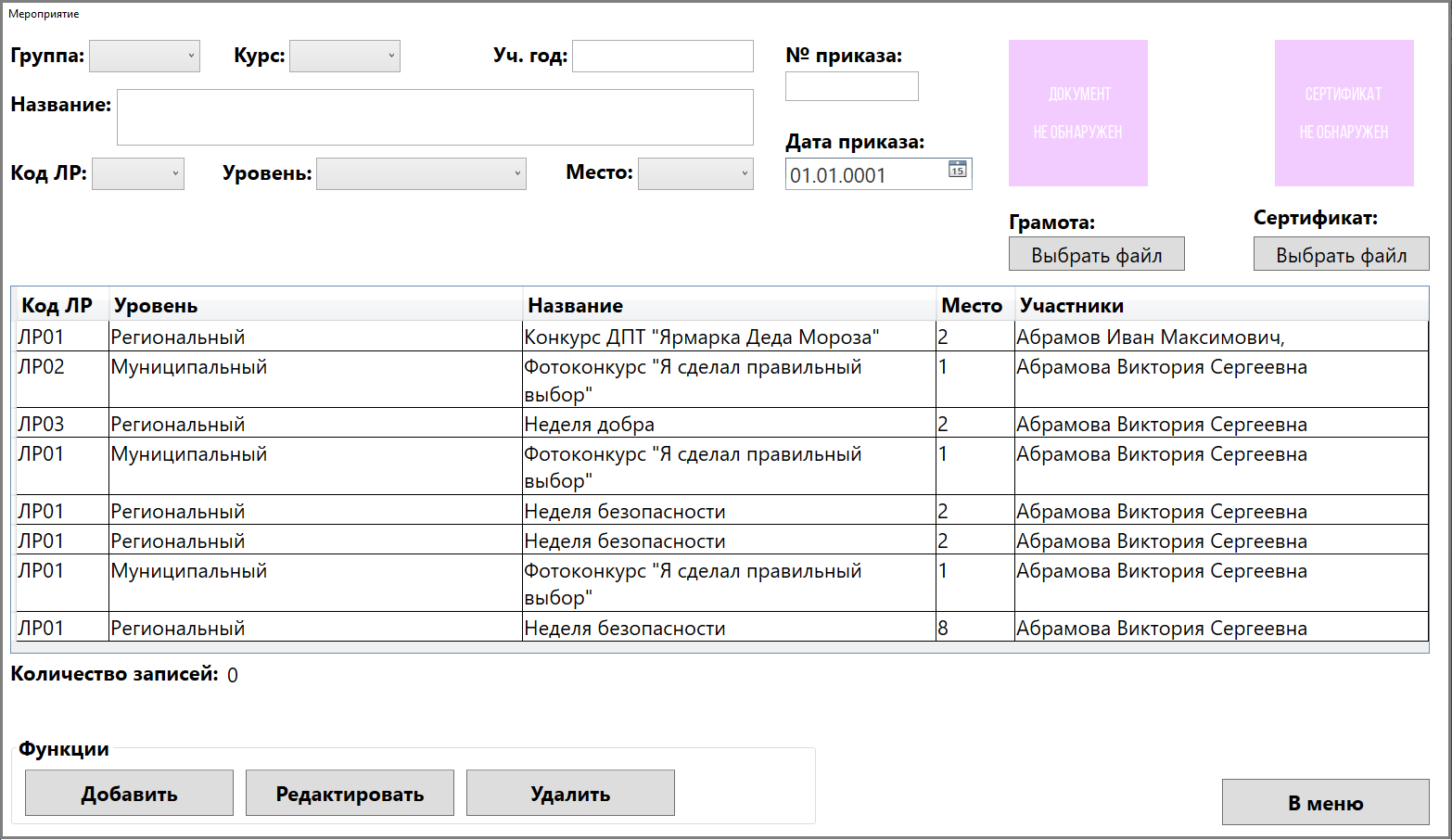


Рисунок Б.2.14 – Окно «Мероприятие»

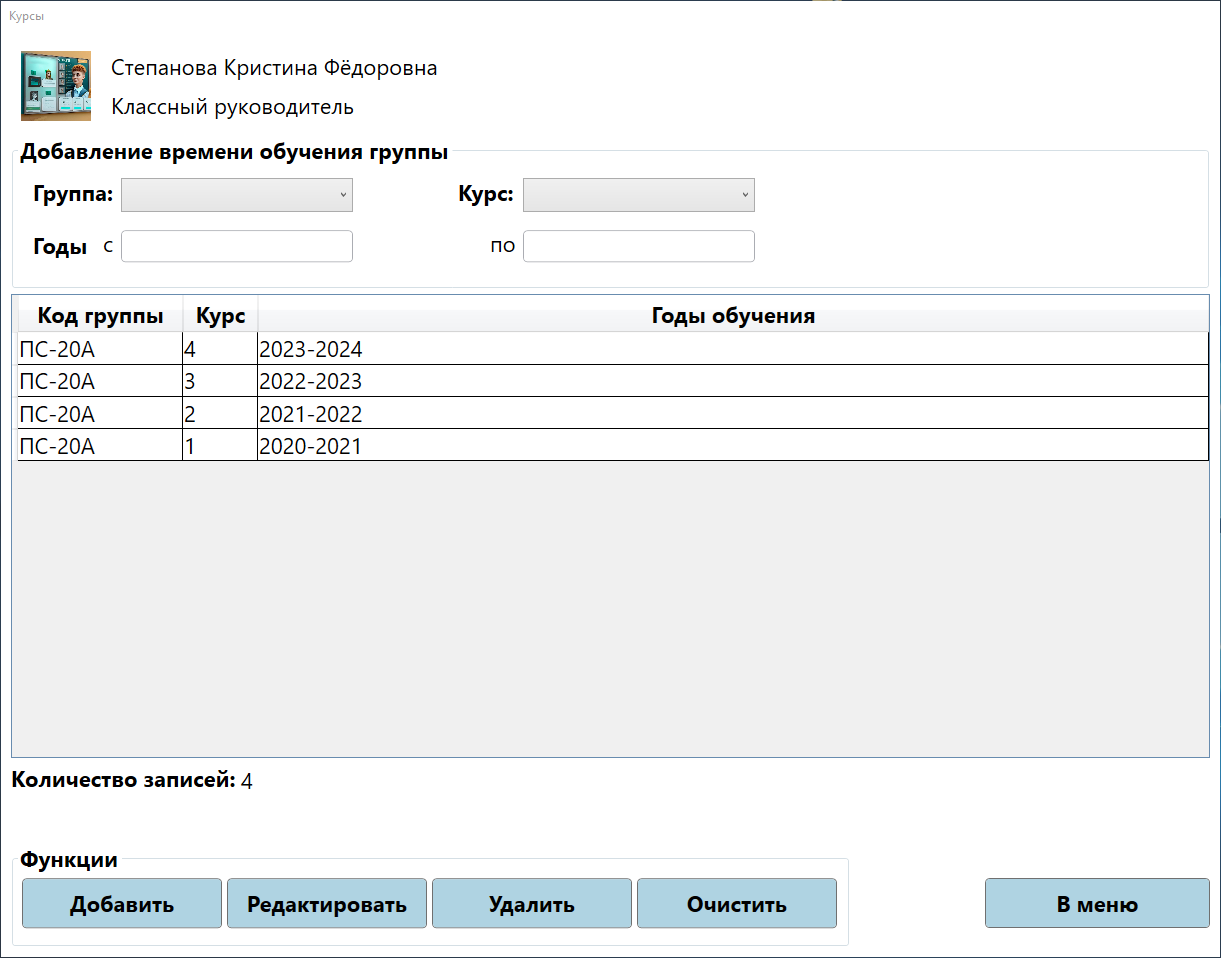


Рисунок Б.2.15 – Окно «Курсы»

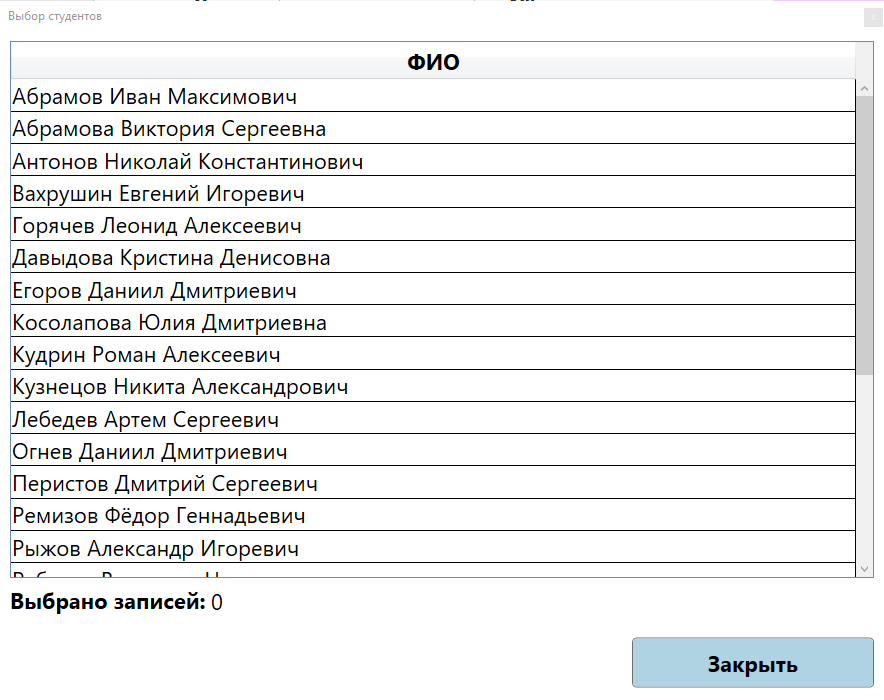


Рисунок Б.2.16 – Модальное окно «Выбор студентов»

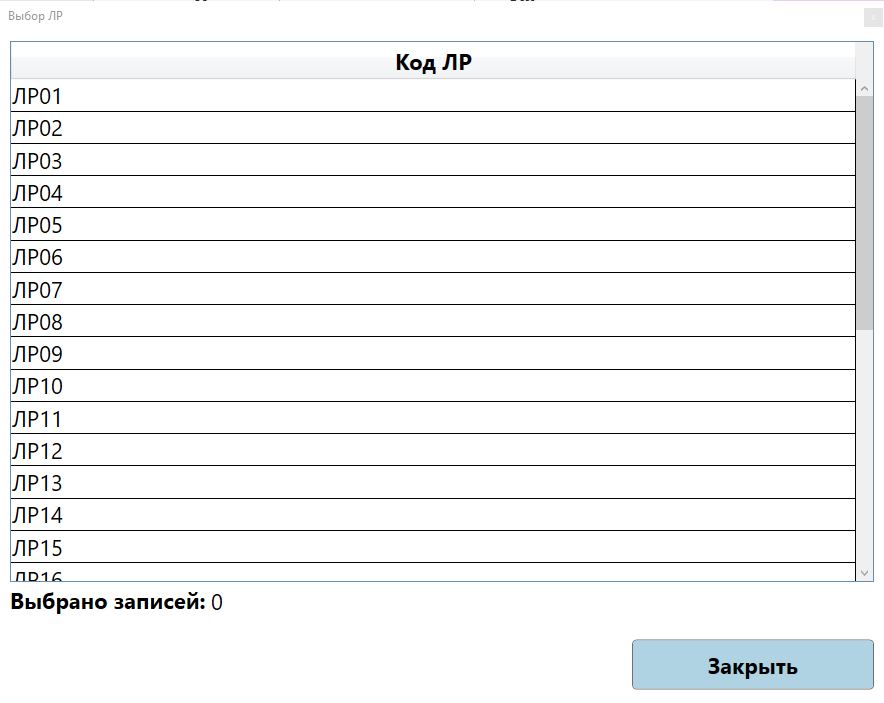


Рисунок Б.2.17 – Модальное окно «Выбор ЛР»

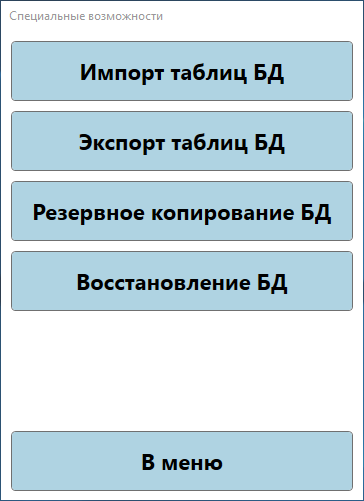


Рисунок Б.2.18 – Окно «Специальные возможности»

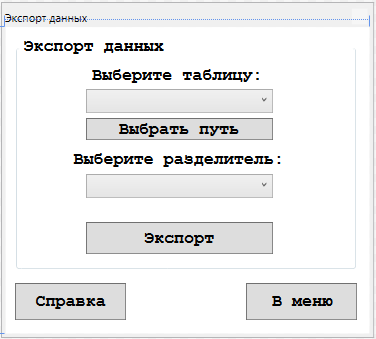


Рисунок Б.2.19 – Окно «Специальные возможности», раздел «Экспорт БД»

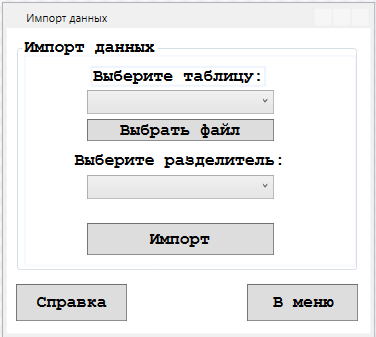


Рисунок Б.2.20 – Окно «Специальные возможности», раздел «ИмпортБД»

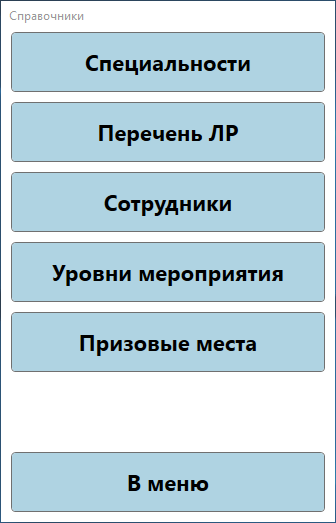


Рисунок Б.2.21 – Окно «Справочники»

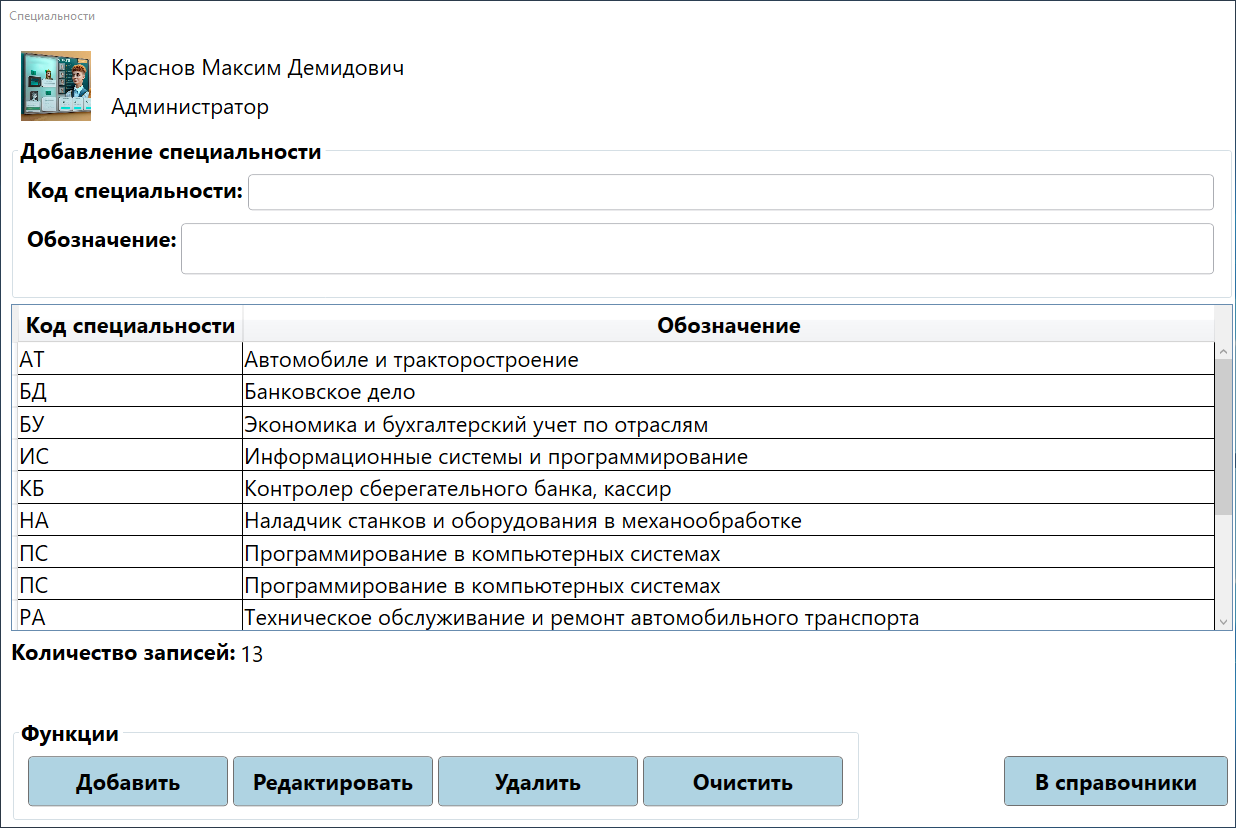


Рисунок Б.2.22 – Окно «Специальности»

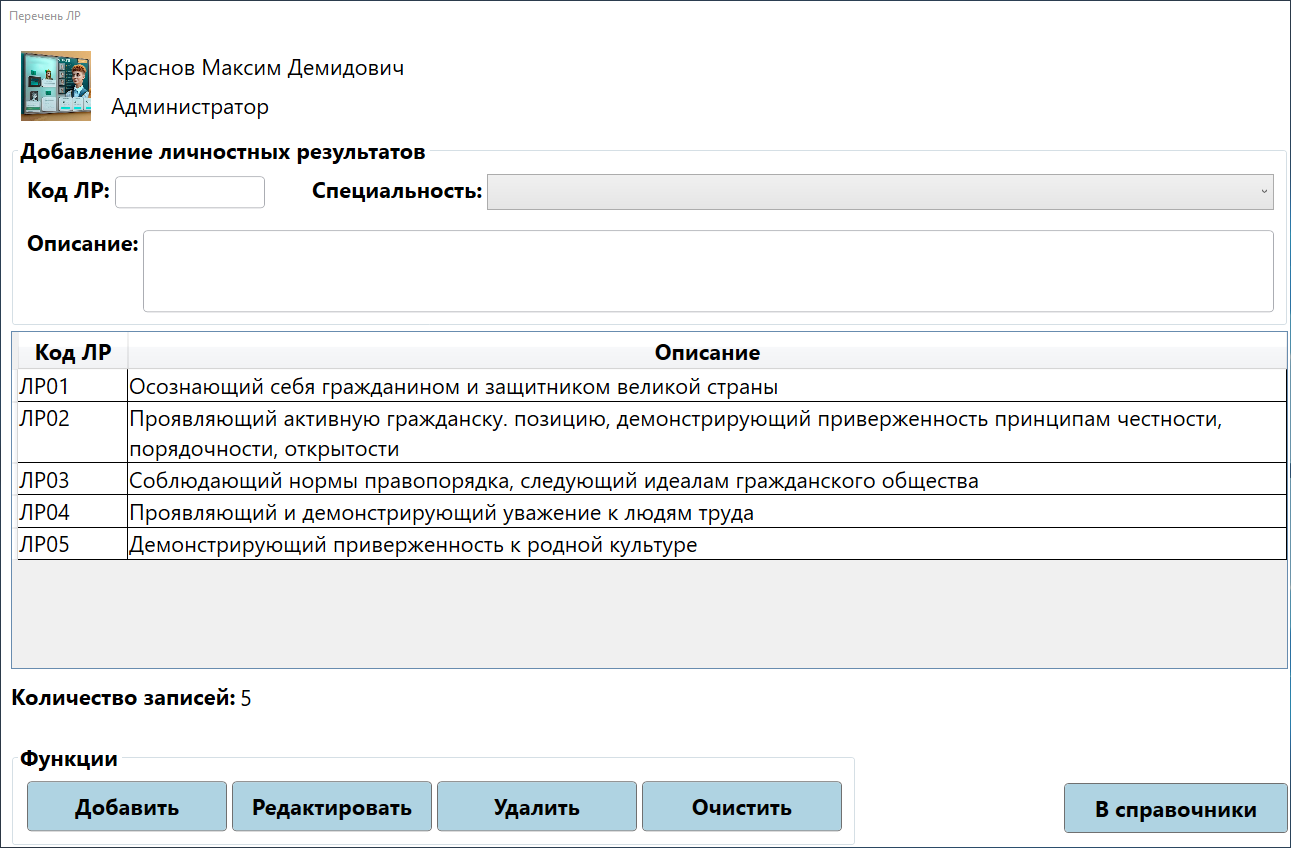


Рисунок Б.2.23 – Окно «Перечень ЛР»

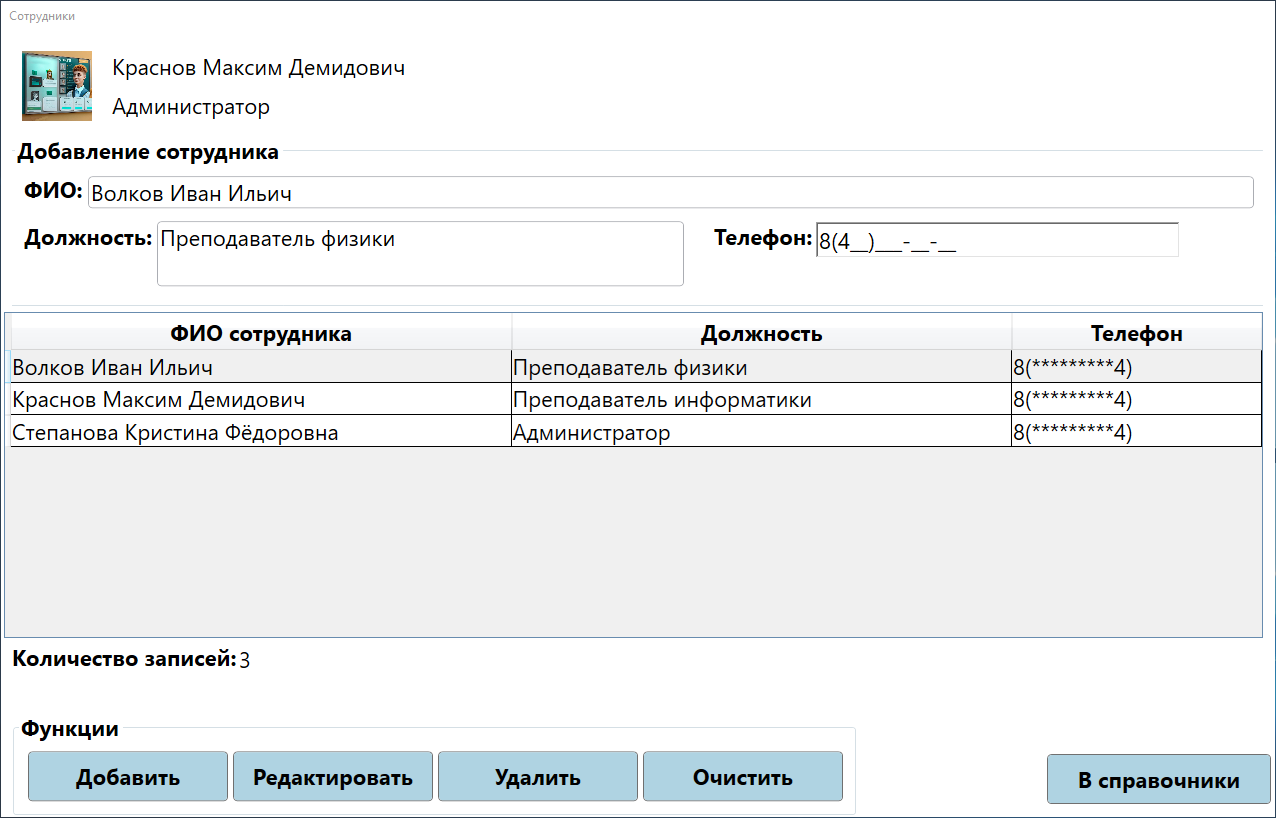


Рисунок Б.2.24 – Окно «Сотрудники»

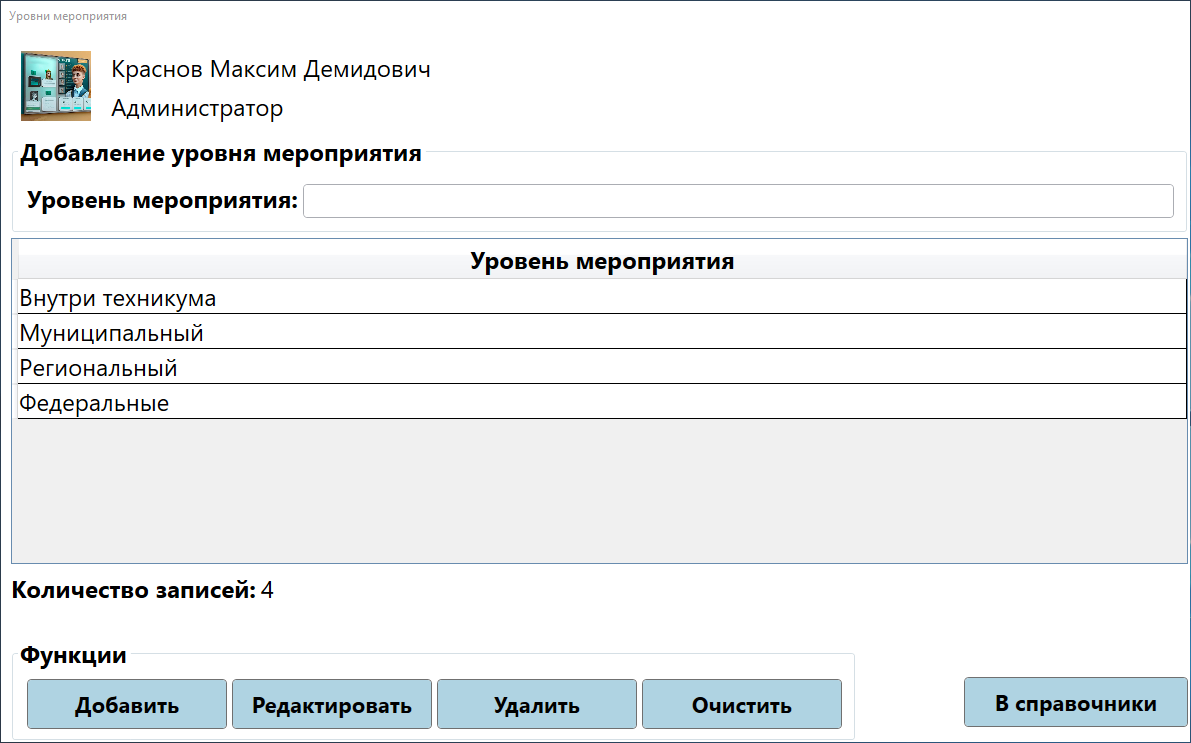


Рисунок Б.2.25 – Окно «Уровни мероприятия»

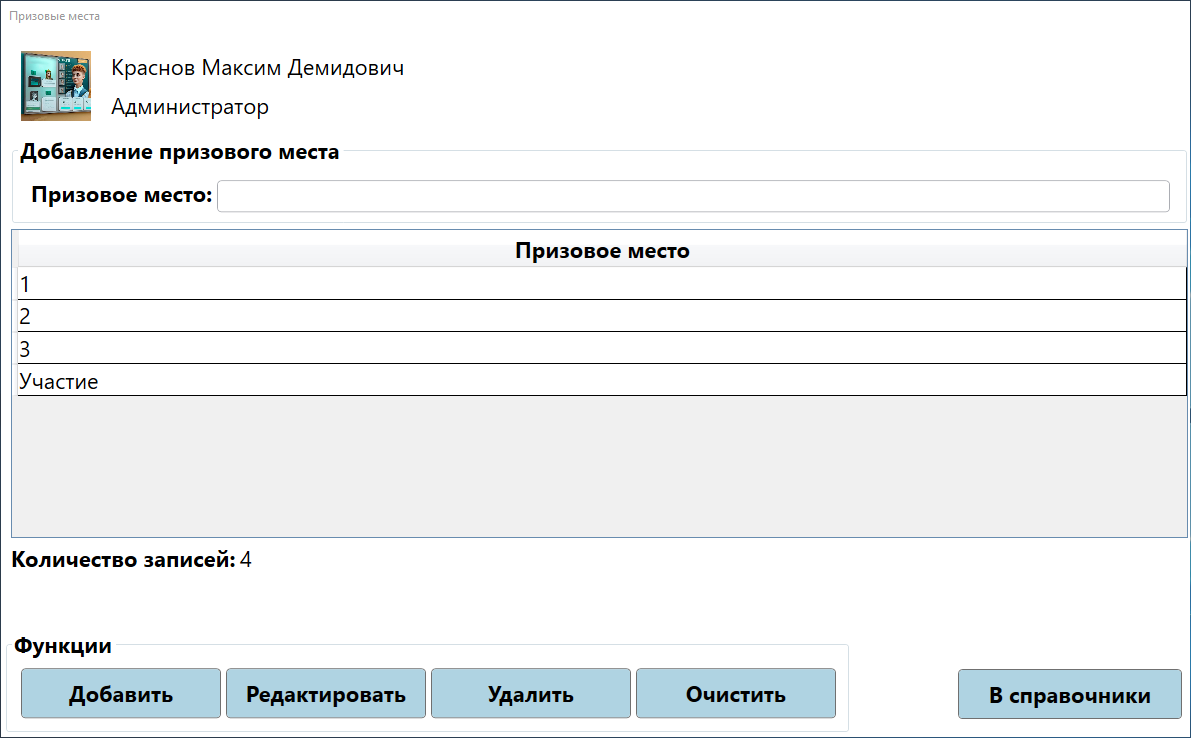


Рисунок Б.2.26 – Окно «Призовые места»

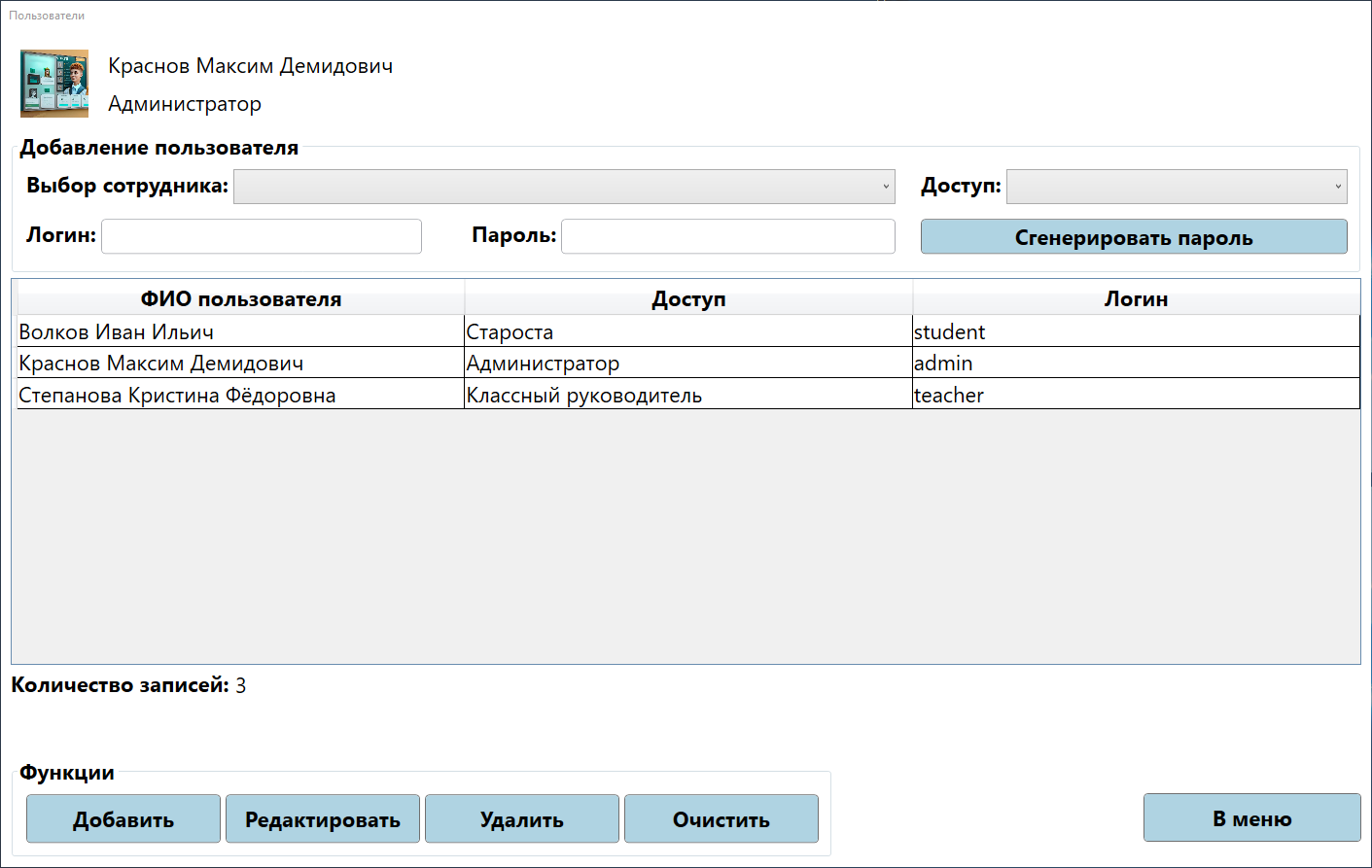


Рисунок Б.2.27 – Окно «Пользователи»

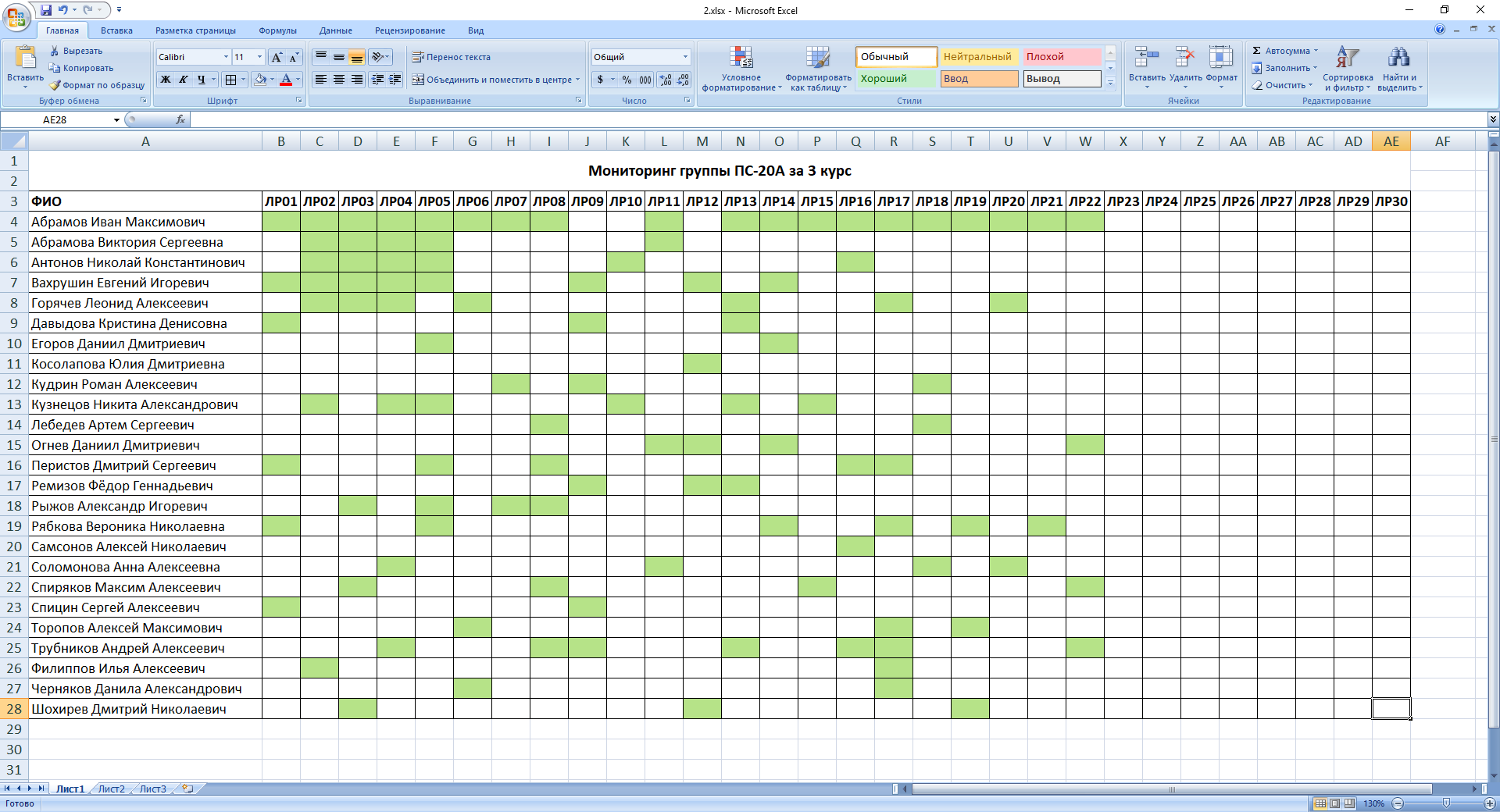


Рисунок Б.2.28 – Прототип отчета «Мониторинг группы» в Excel

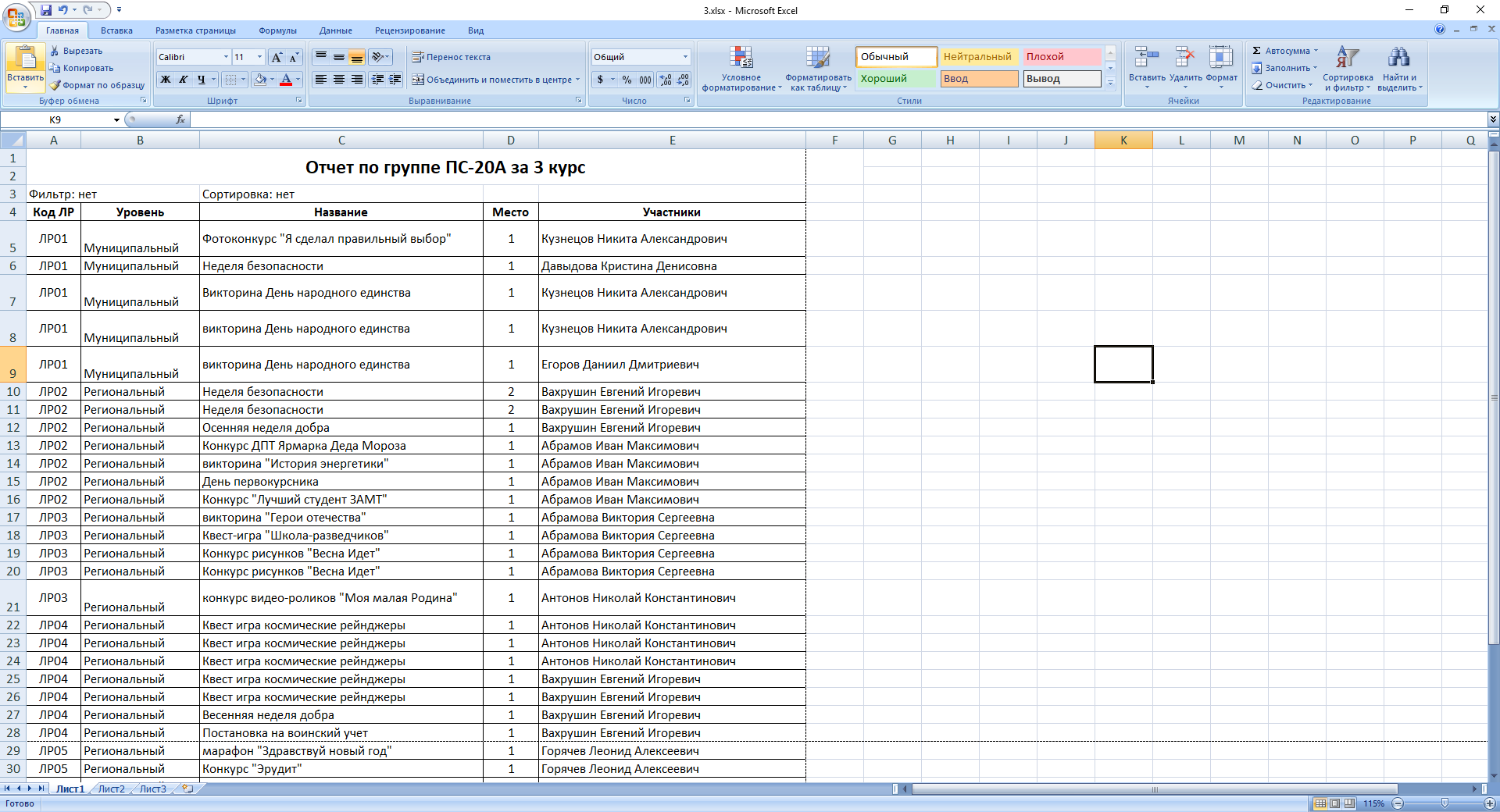
****

Рисунок Б.2.29 – Прототип отчета «Учет мероприятий» в Excel

2.4.2 Описание основных программных модулей

public Authorization() – инициализация формы авторизации

private void Window\_Loaded(object sender, RoutedEventArgs e) – загрузка окна

private void ButExitProgram\_Click(object sender, RoutedEventArgs e) – выход из программы

private void ButAuth\_Click(object sender, RoutedEventArgs e) – код для формы авторизации

private void cbShowPass\_Checked(object sender, RoutedEventArgs e) – переключатель для показывания пароля

public Captch() – инициализация окна капчи

private void Captch\_Loaded(object sender, RoutedEventArgs e) – загрузка окна капчи

public void GenCaptch() – метод генерации капчи

private void CaptchForm\_Closing(object sender, System.ComponentModel.CancelEventArgs e)–обработчик события закрытия формы капчи

private void ButRefresh\_Click(object sender, RoutedEventArgs e) – кнопка обновления капчи

private async void ButInputCaptch\_Click(object sender, RoutedEventArgs e) – кнопка введения сгенерированной капчи

private void ButMenu\_Click(object sender, RoutedEventArgs e) – кнопка перехода в меню

privatevoidFill\_CbSearch() – заполнение фильтра для поиска

privatevoidFill\_DgvRecordEvent() – заполнение таблицы учета мероприятий

privatevoidFill\_CbGroup() – заполнение фильтра по группе

private void Fill\_CbCource(string groupName) – заполнение фильтра по курсу

private void CbGroup\_SelectionChanged(object sender, SelectionChangedEventArgs e) – выбор группы из выпадающего списка

public void GenerateExcel() – метод генерации отчета в Excel

private void ButExpExcel\_Click(object sender, RoutedEventArgs e) – кнопка вывода отчета в Excel

private void CbGroup\_SelectionChanged(object sender, SelectionChangedEventArgs e) – выбор группы из выпадающего списка

private static DataGridCellGetCell(DataGrid dg, int row, int column) – метод обработки ячеек

private static DataGridRowGetRow(DataGrid dg, int index) – метод обработки строк

private static T GetVisualChild<T>(Visual parent) where T : Visual -

private void ButExportExcel\_Click(object sender, RoutedEventArgs e) – кнопка вывода отчета в Excel

privatevoidButChangeUser\_Click(objectsender, RoutedEventArgse) – кнопка выхода из учетной записи пользователя

private void ButStudent\_Click(object sender, RoutedEventArgs e) – кнопка перехода на форму «Студенты»

public ChoiceStudent(List<string> students) – инициализация формы «Выбор студентов»

private void ButClose\_Click(object sender, RoutedEventArgs e) – кнопка закрытия формы

public void AddStudent() – инициализация метода добавления студентов в строку

private void Window\_Loaded(object sender, RoutedEventArgs e) – метод загрузки формы

private void DgvChoiceStudent\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

private void DgvChoiceStudent\_MouseRightButtonDown(object sender, MouseButtonEventArgs e) – занесение студента в строку подвойному клику

private void UpdateCourceTable() – обновление таблицы Курсы

privatevoidCountRowsDgv() – подсчет количества строк

publicvoidFill\_Cource() – заполнение таблицы «Курсы»

publicvoidFill\_ComboBoxGroup() – заполнение фильтра по группам

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопка добавления записи

private void dgv\_cource\_SelectionChanged(object sender, SelectionChangedEventArgs e) – метод обработчика события выбора покурсу

private void ButEdit\_Click(object sender, RoutedEventArgs e) – кнопка редактирования записи

private void tb\_years\_s\_PreviewTextInput(object sender, TextCompositionEventArgs e) – метод предпросмотра года обучения

private void ButDel\_Click(object sender, RoutedEventArgs e) – кнопка удаления записи

private void ButClear\_Click(object sender, RoutedEventArgs e) – кнопка очистки полей

private void ButMenu\_Click(object sender, RoutedEventArgs e) – кнопка перехода в меню

public void Fill\_Event() – кнопка заполнения таблицы Event

public void Fill\_CbGroup() –фильтр заполнения групп

public void Fill\_CbLevelEvent() – фильтр заполнения по уровня мероприятия

public void Fill\_CbPlaceEvent() – фильтр уровня призового места

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопка добавления записи

private void ButEdit\_Click(object sender, RoutedEventArgs e) – кнопка редактирования записи

private void ButDel\_Click(object sender, RoutedEventArgs e) – кнопка удаления записи

private void DeleteEvent(int eventId) – метод удаления мероприятия

private int GetSelectedEventId() – кнопка получения мероприятия

private int GetEventId() – получение id мероприятия

publicintgroup\_id\_u() – метод получения группы

public int event\_code\_lr\_id\_u() – метод получения кода ЛР

public int event\_level\_event\_id\_u() – метод получения уровня мероприятия

public int event\_prize\_place\_id\_u() – метод получения призового места

private List<string>GetStudentsForGroup(string groupCode) – метод получения списка студентов

private void ButUploadDocument\_Click(object sender, RoutedEventArgs e) – кнопка загрузки фотографии грамоты

public void Fill\_Group() – метод заполнения группы

private void DgvEvent\_SelectionChanged(object sender, SelectionChangedEventArgs e) – метод выбора записи в таблице Event

private void UpdateGroupTable() – метод обновления таблицы «Группы»

public void Fill\_ComboBoxSpecialization()–метод заполнения специализации

public int group\_id\_u() – получение id группы

private void tb\_group\_code\_PreviewTextInput(object sender, TextCompositionEventArgs e) – обработчик события на допуск ввода определенных символов

private void tb\_group\_code\_TextChanged(object sender, TextChangedEventArgs e) - обработчиксобытиянадопусквводаопределенныхсимволовиположениякурсора

private void ButEdit\_Click(object sender, RoutedEventArgs e) – кнопкаредактированиязаписи

private void dgv\_group\_SelectionChanged(object sender, SelectionChangedEventArgs e) – обработчиксобытияпринажатииназаписьвтаблице

publicvoidFill\_Student() – заполнение фильтра студента

publicvoidFill\_Group() – заполнение фильтра группы

publicvoidFill\_Status() – заполнение фильтра статуса

publicvoidFill\_User() – заполнение фильтра пользователя

publicvoidFill\_ComboBoxEmployer() – заполнение фильтра сотрудников

private int GetEmployerId(DB\_Connect connection)–методполучения id сотрудника

private void ButExport\_Click(object sender, RoutedEventArgs e) – кнопкаэкспортаотчета

publicGuide() – метод инициализации формы справочников

publicvoidFill\_Employer() – заполнение списка сотрудников

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопкадобавлениязаписи

private void dgv\_employer\_SelectionChanged(object sender, SelectionChangedEventArgs e) – методобработчикасобытияпонажатиюназаписьвтаблице

privatevoidFillLevelEvent() – заполнение выпадающего списка уровня мероприятий

privatevoiddgv\_level\_event\_SelectionChanged(objectsender, SelectionChangedEventArgse) - метод обработчика события по нажатию на запись в таблице

privatevoidFillLevelEventTable() - заполнение выпадающего списка уровня мероприятий в таблице

publicvoidFill\_List\_Result() - заполнение выпадающего списка кода ЛР

publicvoidFill\_ComboBoxSpecialization() - заполнение выпадающего списка специализаций

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопкадобавлениязаписи

private void ButEdit\_Click(object sender, RoutedEventArgs e) - кнопкаредактированиязаписи

privatevoidFill\_ListResultNew() – заполнение таблицы кода ЛР повторно, чтобы вывести запись на первую строку

private void dgv\_list\_result\_SelectionChanged(object sender, SelectionChangedEventArgs e) - методобработчикасобытияпонажатиюназаписьвтаблице

publicvoidFill\_Prize\_Place() – метод заполнения призового места

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопкадобавлениязаписи

private void tb\_place\_name\_TextChanged(object sender, TextChangedEventArgs e) - обработчиксобытиянадопусквводаопределенныхсимволовиположениякурсора

publicvoidFill\_Specialization() – заполнение фильтра специализации

private void ButAdd\_Click(object sender, RoutedEventArgs e) – кнопкадобавлениязаписи

2.5 Применяемые меры защиты информации

Защита данных – это предупреждение несанкцианированного(случайного или умышленного) доступа к данным, их изменения или разрушения данных при сбоях аппаратных или программных средств и ошибках в работе сотрудников группы эксплуатации.

Функции:

* обеспечение безопасности данных;
* обеспечение секретности данных.

В процессе разработки информационной системы были реализованы следующие меры защиты информации:

*1. парольная защита базы данных.*

В соответствии с рисунком 2.21, парольная защита необходима для обеспечения ее защиты от несанкционированного доступа в соответствии с рисунком 2.31.

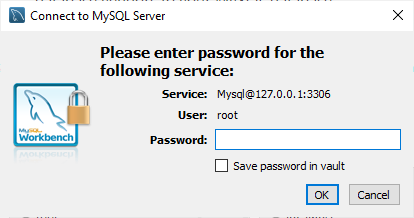


Рисунок 2.21 - Парольная защита базы данных

2. *установки прав доступа,* для того чтобы разграничить доступ пользователям к ресурсам базы. Для этого необходимо поддерживать два фундаментальных принципа: проверку полномочий и проверку подлинности(аутентификацию). Добавлены дополнительные степени защиты: наличие capchaпри неудачном вводе логина и пароля и блокировка на 10 секунд при повторной неудачной попытке,в соответствии с рисунками 2.17 – 2..

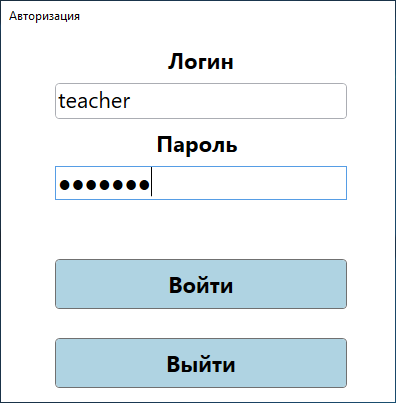


Рисунок 2.17 – Парольная защита программы

Проверка полномочий основана на том, что каждому пользователю или процессу информационной системы соответствует набор действий, которые он может выполнять по отношению к определенным объектам. Проверка подлинности означает достоверное подтверждение того, что пользователь или процесс, пытающийся выполнить санкционированное действие, действительно тот, за кого он себя выдает.

3. *скрытие персональных данных на форме –* одно из направлений, на которое обращает внимание закон «О защите персональных данных» и обращение президента РФ. Применено скрытие данных в таблице и полях ввода на форме Клиенты в соотвествии с рисунком 2.18.

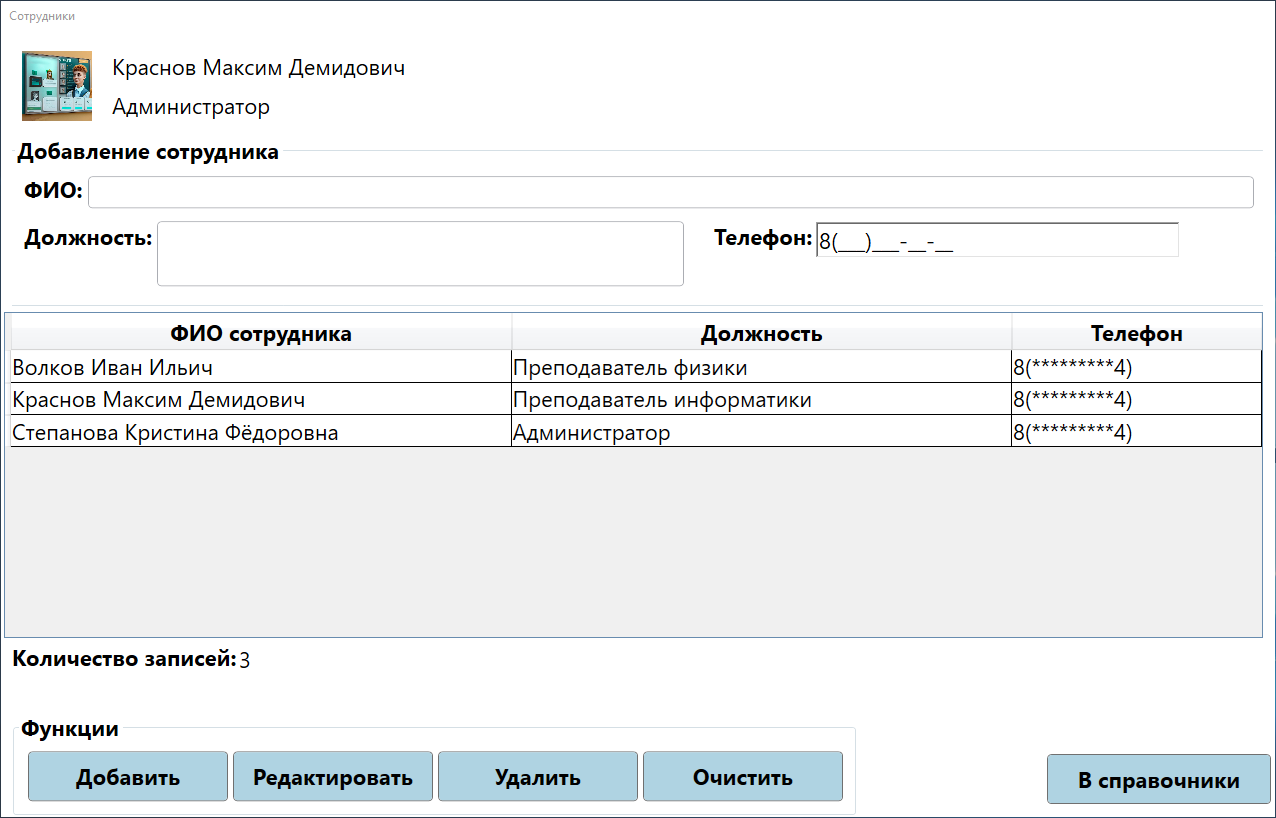


Рисунок 2.18 – Окно «Клиенты»

4. *скрытие форм* при отсутствии активности работы с АИС в течении 3-х минут. Этот способ обезопасит от несанкционированного доступа при оставлении АИС без присмотра в соотвествии с рисунком 2. 19.

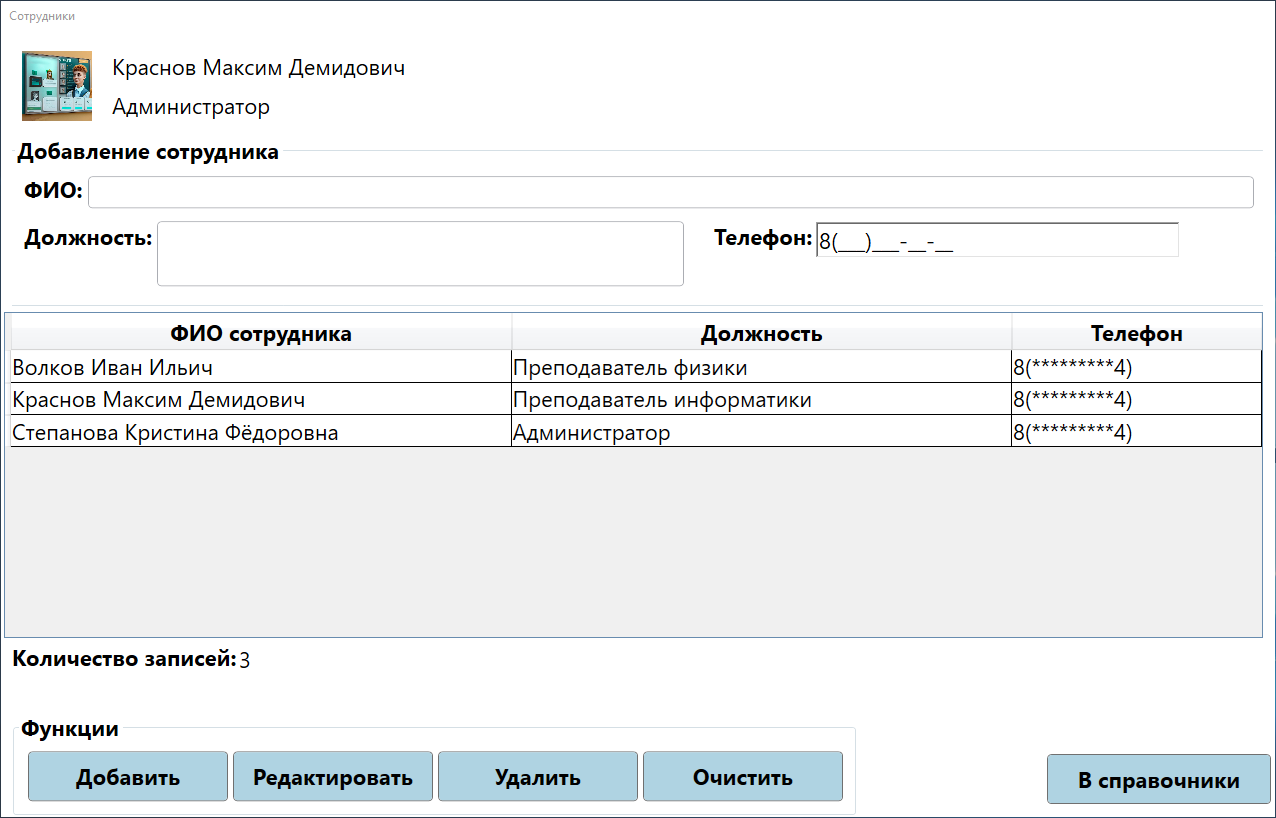
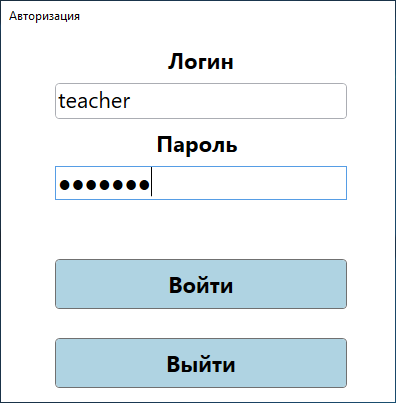


Рисунок 2.19 – Блокировка системы при отсутствии активности

2 *резервное копирование БД -* необходимо для зашиты базы данных от системных сбоев, ошибок и для возможности восстановления продолжения работы с базой данных. Резервное копирование осуществляется автоматически при выходе из программы.

Согласно рисунку 2.20 резервное копирования осуществляется в отдельном меню «Специальные возможности».

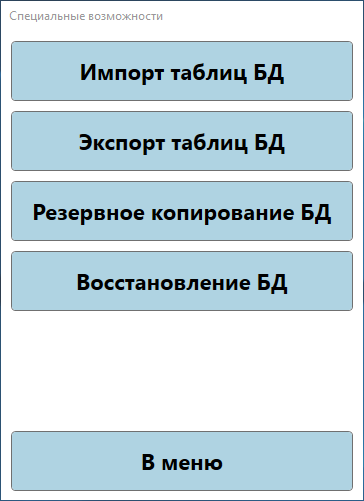


Рисунок 2.20 – Окно «Специальные озможности»

Диалоговое окно восстановлении базы данных, в соответствии с рисунком 2.21

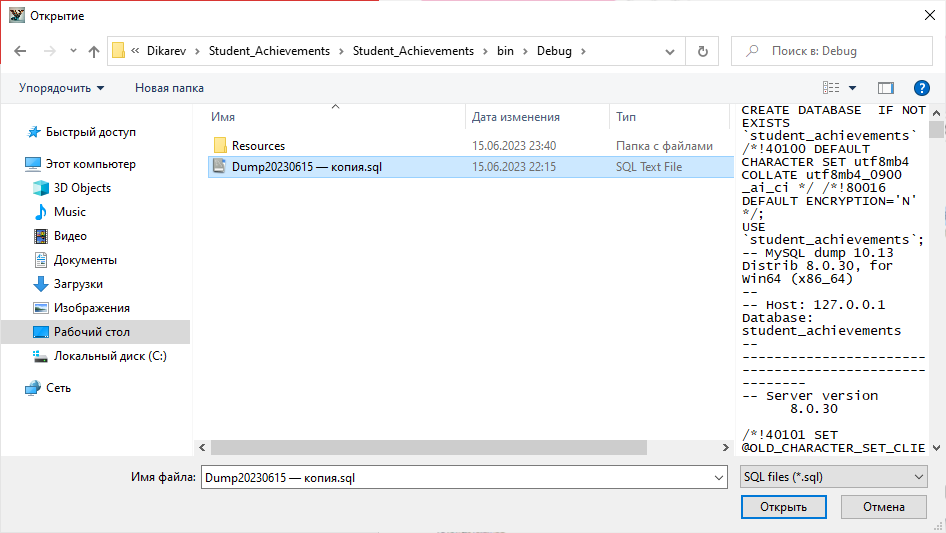


Рисунок 2.21 – Диалоговое окно восстановления базы

Существует несколько причин, при которых база данных может оказаться поврежденной.

Восстановление после сбоев осуществляется при нажатии кнопки «Восстановить базу», откроется диалоговое окно, где нужно выбрать необходимую резервную копию.

**3 Тестирование**

Тестирование – это процесс, состоящий из ряда мероприятий, направленных на проверку объекта тестирования, с целью определения соответствия его характеристик заданным требованиям и выявления возможных дефектов. Дефектами могут быть как ошибки в работе, так и неприемлемое качество функционирования в определенных условиях эксплуатации.

Результаты тестирования представляют собой список обнаруженных дефектов, но, как правило, без указания причин их возникновения. Тестирование программного продукта проводят на различных этапах его создания, а также в процессе технического сопровождения и дальнейшего развития.

Этот процесс является одним из наиболее эффективных способов обеспечения качества программного обеспечения и является неотъемлемой частью системы обеспечения качества продукта.

Таблица 3.1 – Тестирование программы

|  |  |  |
| --- | --- | --- |
| Название элемента формы | Ожидаемый результат | Полученный результат |
| 1 | 2 | 3 |
| **Окно авторизации** | | |
| Заполнение поля логин | Блокировка ввода русского алфавита | Корректное выполнение команды |
| Заполнение поля пароль | Блокировка ввода русского алфавита | Корректное выполнение команды |
| Кнопка «Войти» | При правильно веденных логине и пароле – переход в главное меню | Корректное выполнение команды |
| Кнопка «Войти» | При двухкратной ошибке ввода логина или пароля открытие модального окна с capcha | Корректное выполнение команды |
| Кнопка «Войти» | При ошибке логина или пароля на форме авторизации с capcha  задержка на 10 секунд | Корректное выполнение команды |
| Кнопка «Выйти» | Завершение работы приложения | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Окно главного меню администратора** | | |
| Кнопка «Справочники» | Переход на форму справочники | Корректное выполнение команды |
| Кнопка «Пользователи» | Переход на форму пользователи | Корректное выполнение команды |
| Кнопка «Специальные возможности» | Переход на форму специальных возможностей | Корректное выполнение команды |
| Кнопка «Выход из учетной записи» | Выход из учетной записи пользователя | Корректное выполнение команды |
| **Окно главного меню студента** | | |
| Кнопка «Группа» | Переход на форму группа | Корректное выполнение команды |
| Кнопка «Список студентов» | Переход на форму список студентов | Корректное выполнение команды |
| Кнопка «Курсы» | Переход на форму курсы | Корректное выполнение команды |
| Кнопка «Мероприятия» | Выход из аккаунта с переходом на форму мероприятия | Корректное выполнение команды |
| Кнопка «Выход из учетной записи» | Выход из учетной записи пользователя | Корректное выполнение команды |
| **Окно главного меню классного руководителя** | | |
| Кнопка «Группа» | Переход на форму группа | Корректное выполнение команды |
| Кнопка «Список студентов» | Переход на форму список студентов | Корректное выполнение команды |
| Кнопка «Курсы» | Переход на окнокурсы | Корректное выполнение команды |
| Кнопка «Мероприятия» | Переход на форму мероприятия | Корректное выполнение команды |
| Кнопка «Отчеты» | Переход на форму отчеты | Корректное выполнение команды |
| Кнопка «Выход из учетной записи» | Выход из учетной записи пользователя | Корректное выполнение команды |
| **Окно Справочники** | | |
| Кнопка «Специальности» | Переход на форму специальностей | Корректное выполнение команды |
| Кнопка «Перечень ЛР» | Переход на форму переченя ЛР | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Сотрудники» | Переход на форму сотрудников | Корректное выполнение команды |
| Кнопка «Уровни мероприятия» | Переход на форму уровней мероприятий | Корректное выполнение команды |
| Кнопка «Призовые места» | Переход на форму призовых мест | Корректное выполнение команды |
| Кнопка «В меню» | Переход на форму главного меню | Корректное выполнение команды |
| **Окно Специальности** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех соответствующих полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Перечень ЛР** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Кнопка «Добавить» |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Кнопка «Редактировать» |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Кнопка «Удалить» |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Очистить» | Очистка всех соответствующих полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Сотрудники** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех соответствующих полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Уровни мероприятия** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| **Окно Призовые места** | | |
| Кнопка «Добавить» | При вводе полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Пользователи** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи в таблице производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| Кнопка «Сгенерировать пароль» | Генерация пароля для поля «Пароль» на форме | Корректное выполнение команды |
| **Окно Специальные возможности** | | |
| Кнопка «Импорт таблиц» | Диалоговое окно Windows для импорта папки csv | Корректное выполнение команды |
| Кнопка «Экспорт таблиц» | Диалоговое окно Windows для экспорта таблиц | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Резервное копирование БД» | Блокировка нежелательного ввода | Корректное выполнение команды |
| Кнопка «Восстановление БД» | блокировка нежелательного ввода, добавление шифровния | Корректное выполнение команды |
| **Окно Группы** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Список студентов** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Окно Курсы** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| **Окно Мероприятия** | | |
| Кнопка «Добавить» | При вводе соответствующих полей производить заполнение в таблицу | Корректное выполнение команды |
| Кнопка «Редактировать» | При выборе записи по клику мыши в таблице после изменения данных производить редактирование записи | Корректное выполнение команды |
| Кнопка «Удалить» | При выборе записи по клику мыши в таблице после нажатия на кнопку | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В меню» | Переход в меню | Корректное выполнение команды |
| Кнопка «Выбрать файл» | Открытие диалогового окна Windowsдля загрузки файла | Корректное выполнение команды |
| **Окно Отчеты** | | |
| Кнопка «Учет мероприятий» | Переход на форму «Учет мероприятий | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Мониторинг группы» | Переход на форму «Мониторинг группы | Корректное выполнение команды |
| **ФормаУчет мероприятий** | | |
| Кнопка «Отчет в Excel» | Открытие Excelи печать информации с таблицы DataGridView | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В отчеты» | Возвращение на окно «Отчеты» | Корректное выполнение команды |
| **ФормаМониторинг группы** | | |
| Кнопка «Отчет в Excel» | Открытие Excelи печать информации с таблицы DataGridView | Корректное выполнение команды |
| Кнопка «Очистить» | Очистка всех полей на форме | Корректное выполнение команды |
| Кнопка «В отчеты» | Возвращение на окно «Отчеты» | Корректное выполнение команды |

**4 Экономическое обоснование**

Разрабатываемое программное приложение предназначено для организации учета личностных достижений студентов техникума и мониторинга личностных результатов за курс и весь период обучения. Существующие аналоги программного продукта имеют ряд недостатков:

* являются платными и требуют предварительную настройку;
* выполняют лишь ряд функций – например портфолио;
* являются частью АИС для обеспечения образовательного процесса в школе и не подходят для конкретного заведения.

В рамках выпускной квалификационной работы было выбрано одно направление – «Учет заказов». Реализации данного направления достаточно для демонстрации профессиональных навыков программирования в области автоматизации деятельности менеджера и администратора системы при работе с клиентами на платформе .NET.

Разрабатываемая АИС позволит автоматизировать процессы:

* + учет участия обучающихся группы в различных мероприятиях;
  + просмотр журнала учета мероприятий с использованием поиска, сортировки, условного форматирования и различных фильтров;
  + вести мониторинг личностных результатов студентов по курсам и периоду обучения;
  + формирование отчета по мероприятиям группы за курс и период обучения;
  + формировать отчет по достижениям конкретного студента группы за курс и период обучения;
  + экспортировать отчеты в приложения MS Excel 2007
  + выполнять импорт и экспорт данных, резервное копирование и восстановление БД.

Программное приложение позволит:

* оперативно осуществлять выборку данных из базы;
* редактировать базу данных.

Стадии разработки программного продукта представлены в таблице 4.1

Таблица 4.1 – Стадии и этапы разработки

| № п/п | Название этапа | Содержание | Кол-во дней |
| --- | --- | --- | --- |
| 1 | Анализ | Анализ предметной области | 2 |
| 2 | Проектирование | Составление технического задания | 3 |
| 3 | Разработка | Разработка структуры БД | 26 |
| Разработка форм программного приложения |
| Разработка программного кода |
| 4 | Тестирование | Проверка работоспособности всех функций программного продукта | 1 |
| 5 | Документирование | Составление руководства пользователя | 3 |
| Составление листинга | 1 |
| Составление пояснительной записки | 5 |

На изготовление программного продукта будет потрачен 40 дней.

Приведенные в данном разделе выпускной квалификационной дипломной работы результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные данные для расчета экономических показателей приведены в таблице 4.2.

Таблица 4.2 Исходные данные для расчета экономических показателей

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Наименование показателя | Единицы измерения | Значение показателя |
| СЭВМ | Стоимость ЭВМ | руб. | 42000 |
| ДМ | Среднее количество рабочих дней в месяце | дни | 24 |
| Счас | Стоимость час работы программиста | руб. | 100 |
| ЦЭЛ | Тариф за 1 кВт/час | руб. | 4,48 |
| Тчас | Количество часов работы над разработкой в день | час | 5 |
| P | Мощность, потребляемая ЭВМ | кВт | 0,7 |
| Тинтер | Тариф за услугу Интернет, месяц | руб. | 320 |

4.1 Расчет себестоимости программного продукта

Себестоимость разработки программного продукта рассчитываются по следующей формуле:

|  |  |
| --- | --- |
| Спп= ЗП + Зэл+ А + Зпр | (1) |

где ЗП– затрата программиста;

Зэл – затраты на электроэнергию;

А – амортизация оборудования;

Зпр– прочие расходы.

Зарплата программиста рассчитывается по формуле:

|  |  |
| --- | --- |
| ЗП **=**Сч**×**Тчас**×** Д | (2) |

где Сч– стоимость час работы программиста (руб.);

Тчас– количество часов работы над разработкой в день;

Д– количество дней затраченных на разработку программного продукта.

Согласно таблице 4.1 на разработку программного продукта ушло 24 рабочих дня. Среднее количество часов работы в день составило 5 часов. Так как навыков разработки программных продуктов мало, то часовая ставка начинающего программиста составит 100 рублей, что соответствует данным таблицы 4.2

|  |  |
| --- | --- |
| ЗП **=** 100 **×**5**×**24 = 12000 руб. |  |

Затраты на электроэнергию рассчитывается по формуле:

|  |  |
| --- | --- |
| Зэл**=**Чотраб**×** Ц1кВт**×** Р | (3) |

где Чотраб– количество часов работы оборудования (ПК), освещения;

Ц1кВт – стоимость 1кВт электроэнергии, (руб.);

Р – мощность, потребляемая ЭВМ, (кВт).

Количество часов работы оборудования (ПК), освещения рассчитывается по формуле:

|  |  |
| --- | --- |
| Чотраб**=**Тчас**×** Д | (4) |

где Тчас– количество часов работы над разработкой в день;

Д– количество дней затраченных на разработку программного продукта.

Количество дней на разработку составило 40, что соответствует данным таблицы 4.1. Количество часов работы в день 5, в соответствии с таблицей 4.2

|  |  |
| --- | --- |
| Чотраб **=**5**×**40 = 200 часов |  |

В соответствии с данными таблицы 4.2 стоимости 1кВт электроэнергии равна 4,48 рублей, а потребляемая мощность ПК составляет 0,7 кВт в час.

|  |  |
| --- | --- |
| Зэл**=**200**×** 4,48**×** 0,7= 627,20 руб. |  |

Амортизационные отчисления рассчитываются с учетом нормы амортизации (Нам**=** 5-10%):

|  |  |
| --- | --- |
| А **=** Нам**×** СЭВМ | (5) |

где СЭВМ– стоимость оборудования, согласно таблице 4.2.

|  |  |
| --- | --- |
| А **=** 0,05 **×** 42000 = 2100 руб. |  |

Для разработки программного продукта необходимо использовать Интернет, следовательно определим затраты на Интернет, как прочие.

|  |  |
| --- | --- |
| З пр**=** Цинт(1ч)**×** Тчас**×** Д | (6) |

Цена одного часа работы в Интернет рассчитывается по формуле:

|  |  |
| --- | --- |
| Ц инт(1ч) = Цтар :Дмес : 24 | (7) |

Согласно таблице 4.2 стоимость услуги Интернет в месяц составил 320 рублей. Исходя из стоимости тарифа определим стоимость часа работ в Интернет.

|  |  |
| --- | --- |
| Ц инт (1ч) = 320 :24 : 24 = 0,56 руб.  З пр**=** 0,56**×** 5**×** 24 = 67,20 руб. |  |

Выполнив все расчеты, определим себестоимость программного продукта.

|  |  |
| --- | --- |
| Спп= 12000 + 627,20 + 2100 + 67,20 = 14794,40 рублей≈15000 рублей |  |

4.2 Стратегия продвижения программного продукта на рынок

Сегодня самым эффективным способом продажи программного продукта является Интернет.

Наиболее распространёнными средствами реализации программного продукта в сети Интернет являются: демоверсия, реклама, размещение на различных хостингах.

* + демоверсия продукта — предварительная (ограниченная и, как правило, бесплатная) версия продукта. Демоверсии дают представление пользователям о готовящемся продукте и создаются в качестве рекламы и привлечения вниманию к будущему продукту. В этом смысле они приравниваются к трейлерам для кинофильмов. Обычно они являются в некотором смысле не готовой к продаже версией полного продукта. По этим причинам в большинстве случаев они распространяются бесплатно;
  + реклама (баннеры) - один из преобладающих форматов [Интернет-рекламы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%80%D0%B5%D0%BA%D0%BB%D0%B0%D0%BC%D0%B0). Представляет собой графическое изображение, аналогичное рекламному модулю в прессе, но способное содержать анимированные (редко видео-) элементы, а также может являться [гиперссылкой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B0) на сайт рекламодателя или страницу с дополнительной информацией;
  + размещение на специализированных хостингах – также вариантом распространения программного продукта является размещение на специализированных хостингах, которые являются своего рода специализированными магазинами программного обеспечения.

Из выше перечисленных вариантов распространения программного продукта, было выбрано размещение дипломного проекта за сайте prepod24.ru, который взимает плату 20% от стоимости работы. Дипломные работы вирируются в размере от 3000 до 12000 рублей.

4.3 Расчет цены разработанной программы

Цена разработанной программы определяется по следующей формуле:

|  |  |
| --- | --- |
| Ц пп= Спп | (8) |

где Спп – себестоимость программного продукта (руб.);

Зпродв – суммарные затраты на продвижение программы (руб.);

Ц пп= 15000руб

Выручка от продаж при условии  – количество пользователей, желающих прибрести программу, составит:

|  |  |
| --- | --- |
|  | (10) |

Было принято решение продавать информационную систему по ведению базы данных за 4000 рублей. При продаже каждой копии выручка составит 2800 рублей.

Срок окупаемости инвестиционного проекта () – это период времени, который потребуется для возмещения инвестиций. – определяется с учетом дисконтирования, путем суммирования ежегодных поступлений до определенного периода, в котором они превзойдут первоначальные расходы денежных средств. Расчёт срока окупаемости представлен в таблице 4.3

Таблица 4.3Расчет срока окупаемости разработанной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период | Цена ПП,  руб. | Кол-во покупок  шт. | Выручка,  руб. | Накапливаемая выручка,  руб. |
| 1 год | 4000 | 4 | 11200 | 11200 |
| 2 год | 4000 | 2 | 5600 | 16800 |
| Итого | 4000 | 6 | 16800 |  |

4.4 Технико-экономические показатели разработки программы

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 4.3.

Таблица 4.3 – Технико-экономические показатели разработки программы

|  |  |
| --- | --- |
| Показатель | Значение |
| Себестоимость программного продукта (руб.) | 15000 |
| Цена разработки ПП (руб.) | 15000 |
| Отпускная цена (руб.) | 4000 |
| Срок окупаемости (года) | 2 |

Данные произведенные расчеты показывают, что разработанный программный продукт является эффективным с экономической точки зрения.

**Заключение**

В ходе выполнения выпускной квалификационной работы была созданаавтоматизированная инфомационня систем по учету личностных достижений обучающихся, отвечающее поставленным требованиям.

В процессе написния программного кода, использовался язык программирования высокого уровня. Были улучшены навыки по работе с платформой .Net, БД MYSQL, а так же приобретен навык по разработке подобных проектов.

Программное приложение обеспечивается выполнение следующих функций:

* аутентификация пользователя;
* автоматизации ввода исходных данных (выпадающие списки, переключатели, автоматическая смена языка);
* просмотр, редактирование, добавление и удаление записей в БД, в режиме системного администратора;
* просмотр, редактирование, добавление и удаление записей о личностном результате студентов группы в мероприятиях, в режиме классного руководителя и студента;
* мониторинг освоения личностных результатов обучающихся;
* поиск, сортировка и фильтрация записей по различным критериям;
* проверка на корректность ввода данных;
* формирование отчетов для экспорта в приложение Excel 2007;
* резервное копирование и восстановление БД в режиме администратора.

Программа, которая требует минимальные ресурсы и в дальнейшем есть возможность перехода на архитектуру клиент-сервера. Кроме того, благодаря своей архитектуре, программа может в будущем легко быть расширена и доработана, например, перейдя на клиент-серверную версию, что позволит более эффективно управлять данными и обращаться к ним из разных мест. Программа может использоваться в качестве демонстрации функционала для заключения договора с потенциальным покупателем.

**Литература**

1. Эндрю Троелсен., Язык программирования C# и платформа .NET., 2015г.
2. Алексей Васильев., Программирование на C# для начинающих. Основные сведения, 2018 г.
3. Троелсен Эндрю, Джепикс Филипп, Язык программирования C#7 и платформы .NET и .NET Core, 2018 г.
4. Евдокимов П. В.,C# на примерах, 2019 г.
5. Вагнер Билл., Наиболее эффективное программирование на C#. 50 способов улучшения кода, 2017 г.
6. Мюллер Джон Поль, Семпф Билл., C# для чайников, 2019 г.
7. "HighPerformanceMySQL: Optimization, Backups, andReplication" авторов Бэрон Шварц, Питер Зайтц, Вадим Ткаченко (3-е издание, 2012 г.)
8. "Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB" автораРасселлаДж. Тинглмайера (2-е издание, 2015 г.)