ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3

АНАЛИТИЧЕСКАЯ ЧАСТЬ 5

Анализ предметной области 5

Постановка задачи 6

ПРОЕКТНАЯ ЧАСТЬ 8

Проектирование и реализация базы данных системы 8

Проектирование и реализация пользовательского интерфейса системы 9

Структура приложения 10

Реализация пользовательских форм 11

Разработка запросов 15

Разработка отчетов 18

Меню и справка 20

ЗАКЛЮЧЕНИЕ 21

Список литературы 22

Приложение А. Техническое задание на курсовой проект .…..………………...…………...…...23

# ВВЕДЕНИЕ

Информационные системы бухгалтерского и складского учёта брали начало ещё при зарождении самих информационных технологий. Из-за скорости и удобства передачи информации, формализации данных и организации результатов при помощи ЭВМ ещё с 80-х годов происходило повышение эффективности деятельности.

Информационная система - система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382:2015).

Информационная система позволяет в кратчайшие сроки составить отчёт или документацию по отдельному товару, набору деталей, ассортименту магазина и даже всему складскому наполнению.

Информационные системы помимо снижения трудозатрат на составление документа, позволяют сократить сам документооборот вплоть до отмены от него. Некоторые документы могут быть заменены на вывод отчёта в самой программе или сам документ может быть более исчерпывающий.

Из вышеописанного можно сделать вывод: эффективность работы бухгалтерии или складских учётов напрямую зависит от степени развития информационных систем и их применения в деятельности отдельных организаций.

При учёте поставок крайне полезной была бы система автоматизированного учёта, способная сохранять и передавать в необходимые хранилища отчеты о поставке, количествах продукции и прочих деталях.

Целью курсового проекта является разработка системы учёта продукции на складе и нахождение оптимальных решений возможных проблем. Данная система будет способна создавать отчёт о текущем наполнении склада, дефицитной продукции, поставках как прошедших, так и готовящихся, а также учёте прибывшей и убывшей продукции.

Так как склад не занимается продажами, то в системе не будет учёта продаж, цены или прибылей, так как склад занимается только хранением и учётом продукции, которая была поставлена на хранение заинтересованными лицами.

В данном курсовом проекте будет рассмотрена автоматизация работы с учётом продукции на складе. Результатом данной работы будет информационная система учёта продукции в наличии на складе абстрактной компании, а также будут продемонстрированы преимущества использования информационных систем в данной сфере.

Так как каждая организация, занимающаяся розничной или оптовой продажей нецифровых товаров, имеет склад, нужно учесть, что возможно множество нюансов между организациями и самой продукцией.

Предметом исследования является процесс учета продукции в наличии и процесс приёма новых поставок, а также перемещение продукции со склада.

Для достижения поставленной цели необходимо выполнить следующие задачи:

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

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

Склад – это некоторая площадь, чаще всего в помещении, предназначенная для хранения любого рода материальных ценностей компаний или любого другого заинтересованного лица. Сам по себе склад используется исключительно для аккумулирования, приёма и передачи ресурсов. Иные операции производятся уже иными сущностями, магазинами для продажи, компаниями транспортировки для перевозки и т.д.

Материальные ценности – это товары, имущество или любые другие материальные предметы, являющиеся собственностью заинтересованного лица.

Функции склада:

* Снабжение – это основная функция склада. Данная функция состоит в том, что склад имеет главной целью обеспечение и снабжение какого-либо производства
* Связующая – данная функция заключается в том, что склад, помимо хранения материалов для изготовления, может хранить и готовую продукцию, таким образом склад будет связующим звеном между рынком сбыта и производством.
* Переработка – так как некоторая продукция в своём производственном цикле требует некоторого времени хранения для подготовки к последующему изменению или продаже.
* Перетарка – зачастую именно на складе происходит изменение упаковки продукции, если это необходимо владельцам продукции.
* Спекулятивная – данная функция может быть названа защищающей, сохранение товара при падении ценности товара или требование роста поставок при их росте.

Система, разработанная в данной курсовой работе, будет предназначена для отслеживания, текущего количество отдельной продукции или продукции отдельных клиентов склада: добавление новых товаров и исключение товаров убывших. Данная информационная система не будет отслеживать затраты поставку товара или учитывать прибыли от его продажи.

Поставка – совокупность продукции или любых других материальных ценностей, прибывших на хранение от одного заказчика в определённый момент.

Хоть отслеживанием поставок или продажей склад не занимается, так как это не является частью его деятельности, данная система должна отслеживать поступление и отчисление продукции со склада.

Существуют следующие способы поступления-отчисления продукции со склада:

* FIFO (first-in-first-out отгружается продукция, которая прибыла первой)
* LIFO (last-in-first-out отгружается продукция, которая прибыла последней)
* FEFO (first-expire-first-out отгружается продукция с ближайшим к истечению сроком годности)
* HIFO (highest-in-first-out отгружается продукция себестоимость которой, была завышена)
* LOFO (lowest-in-first-out отгружается продукция себестоимость которой, была занижена)

Так как в системе не будет предусмотрен учёт продаж и цены отдельной продукции, не имеет смыла реализация последних двух методов зависящий от цены товаров.

В системе будет возможность выбор расположения товара на складе с учётом его особенностей:

* Хрупкий – данная продукция должна храниться в помещениях с ручной погрузкой
* Хранение в темноте – данная продукция (чаще всего это бытовая химия) негативно реагирует на долгое воздействие света, в следствии чего приходит в негодность раньше
* Влагобоязнь – данная продукция должна храниться в помещениях с определённым уровнем влажности
* Боязнь излучение – любое излучение портит продукцию (к примеру фотоплёнка)
* Запрет хранения штабелями – данное свойство указывает, что продукцию нельзя хранить штабелями

Также будет предусмотрена возможность добавления новых критериев при необходимости.



Рисунок 1 – контекстная функциональная диаграмма по учету продукции на складе.

IDEF0 – функциональная модель, отображающая основной или набор второстепенных бизнес-процессов, а также входные данные, которые должны быть предоставлены для выполнения определённого бизнес-процесса, механизмы, при помощи которых он выполняется (снизу) и контролирующие выполнение (сверху), а также результаты данного процесса.



Рисунок 2 – диаграмма декомпозиции учёта продукции

Диаграмма декомпозиции – это функциональная модель, описывающая подпроцессы основного бизнес-процесса системы.

## Постановка задачи

Автоматизация данных процессов позволит сохранить информацию в базе, в которую вводится данная информация с помощью удобного интерфейса программы.

В данной информационной системе необходимо разработать:

* Базу данных в MsSQL.
* Приложение в среде программирования.
* Разработать интуитивно понятный и удобный интерфейс.

Так же нужно разработать функциональные возможности системы:

* Редактирование и просмотр данных о продукции.
* Поиск и фильтр данных о продукции.
* Редактирование и просмотр данных клиентов.
* Редактирование и просмотр данных поставок на склад и отправлений со склада.
* Выбор места для товаров на складе

Так же нужно разработать отчеты в системе:

* отчет о количестве продукции на складе по определённым критериям.

Более подробный функционал системы смотри в приложение А (см. Техническое задание на ИС).

# ПРОЕКТНАЯ ЧАСТЬ

## Проектирование и реализация базы данных системы

Для корректного построения базы данных, которая будет использоваться в данной системе, будет логично начать с построения модели данных или ER-модели.

ER-модель – модель данных или схема вида «сущность-связь». Данная модель предназначена для отображения сущностей, присутствующих в системе, взятой для моделирования, а также связей между ними.

Так как мы будем использовать реляционную базу данных в нашей системе, все связи будут в виде «один-ко-многим», это значит, что на объект в первой таблице может ссылаться множество объектов из второй, но объект из второй таблицы может ссылаться лишь на один объект из первой.

Реляционная база данных – база данных, объекты которой могут иметь связи с объектами в других таблицах данной базы.



Рисунок 2 – ER-модель БД.

Разработанная ER-модель состоит из 9-и сущностей: транзакция, клиент, список продукции, продукция/условие, условие хранения, место/условие и место хранения.

Сущности “Транзакция”, “Клиент”, “Список продукции”, “Тип транзакции” и “Сотрудник” предназначены для выполнения следующих задач базы данных:

* Хранение контактной информации о клиенте
* Отслеживание текущей продукции на складе
* Отслеживание транзакций склада, то есть не только получение продукции, но и её отправка
* Отслеживание даты поступления продукции, так как предполагается использовать разные способы поступления-отчисления, в том числе зависящие от даты поставки
* Также присутствует необязательное поле срока годности, необязательное оно так как предполагается, что склад имеет возможность хранить совершенно разные виды продукции

Сущности “Условие хранения”, “Место хранения”, “Место/условие” “Продукция/условие”, предназначены для реализации возможности учёта расположения продукции на складе в зависимости от особенностей хранения продукции.

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

На рисунке 3 представлена физическая схема БД, реализованная на платформе MS SQL Server 2017. На модели отражены названия атрибутов и их типы данных на сервере. Так же на модели отображены связи между сущностями для обеспечения целостности данных.

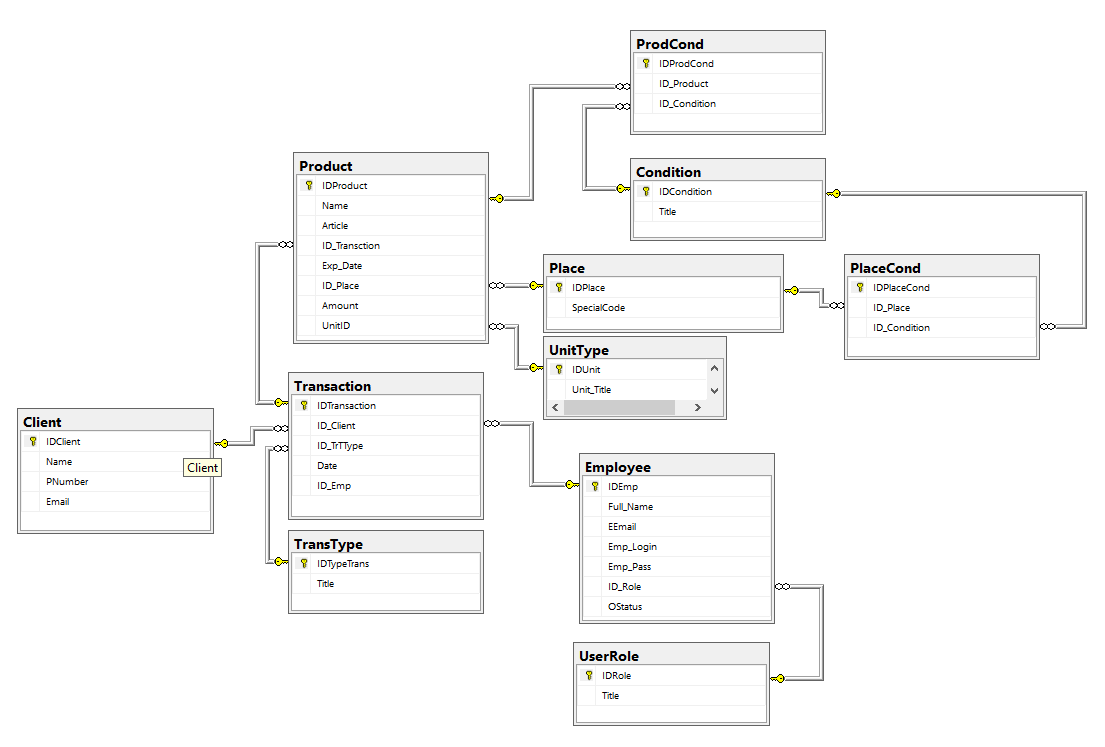


Рисунок 3 – физическая модель БД.

## Проектирование и реализация пользовательского интерфейса системы

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

Visual Studio - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют, разрабатывать как консольные приложения, так и приложения с графическим интерфейсом. В данном проекте будет использована технология WPF.

WPF – система для построения пользовательского интерфейса в основе которой находится векторная система визуализации, что даёт возможность не зависеть от разрешения устройства на котором используется данный интерфейс.

Преимущества Visual Studio по сравнению с аналогичными программными продуктами это:

* высокая популярность позволяет находить решение проблем быстрее;
* низкие требования разработанного приложения к ресурсам компьютера;
* удачная проработка иерархии объектов;
* частично бесплатная;
* удобна в использовании.

Так же был использован язык программирования C# так как:

* также имеет высокую популярность и поддержку Майкрософт;
* в последнее время много совершенствуется;
* высокое удобство написания кода, что уменьшает возможное количество ошибок;
* прост, но имеет высокую эффективность и перспективы в развитии;

## Структура приложения

Для реализации поставленных задач была выбрана клиент-серверная архитектура приложения.

Клиент-серверная архитектура — это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент-сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине.

Данная архитектура обладает рядом преимуществ:

* Отсутствие дублирования кода программы-сервера программами-клиентами.
* Так как поиск данных выполняется на сервере, путём отправки запроса к БД, то требования к компьютерам, на которых установлен клиент, снижаются.
* Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов.
* На сервере проще организовать контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Сервер БД

Клиент

Рисунок 4 – Схема клиент-серверного приложения

Разработанное клиентское приложение выполняет все поставленные задачи, и имеет сложную архитектуру, представленную на Рисунке 5.



Рисунок 5 – схема приложения

На схеме разными цветами выделены разграничения прав доступа. Зелёным цветом выделены функции, доступные над-администратору, красным и синим цветом отмечены функции доступные администратору и пользователю соответственно, черным выделены функции доступные для всех.

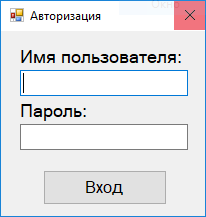
## Реализация пользовательских форм

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

MDI интерфейс – это способ организации графического интерфейса пользователя, предполагающий использование оконного интерфейса, в котором большинство окон расположены внутри одного общего окна.

В начале приложения было разработано девять основных форм, а также дополнительные формы. Первой формой была разработана авторизации для нескольких пользователей. На Рисунке 6 представлена форма авторизации.

Рисунок 6 – Форма авторизации



Авторизоваться можно с 2 пользователей и 1 админа. У пользователей ограниченный доступ к 2 отчетам и функциям добавлениям пользователей. Пример ограничений доступом можно рассмотреть на рисунках 7 Возможности Администратора и 8 Возможности пользователя.

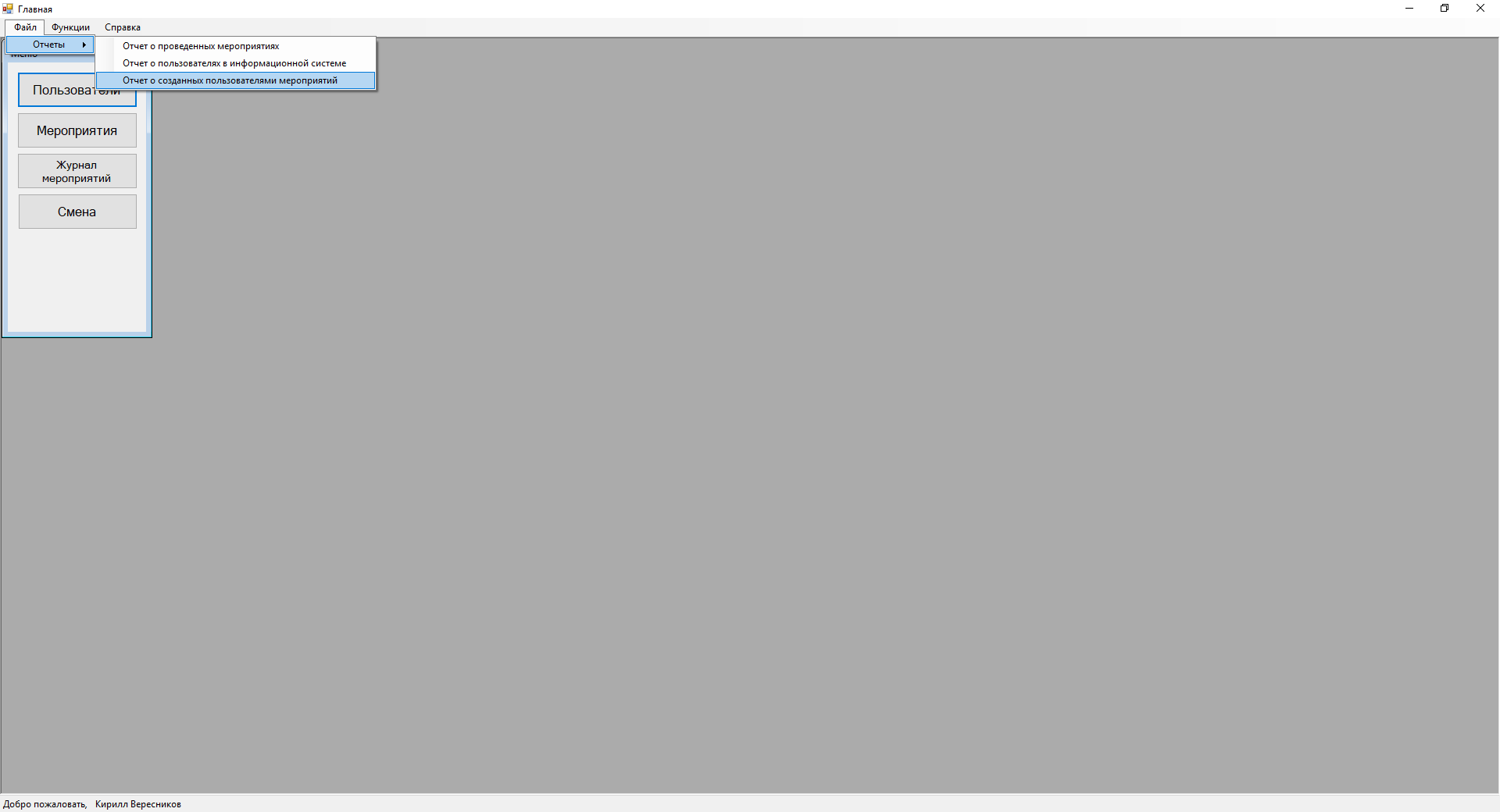


Рисунок 7 – Возможности администратора.



Рисунок 8 – Возможности пользователя.

Код авторизации:

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text != String.Empty && textBox2.Text != String.Empty)

{

if (auth(textBox1.Text, textBox2.Text))

{

if (searchForm("mainForm") == false)

{

mainForm = new mainForm();

mainForm.Show();

}

mainForm = new mainForm();

mainForm.refreshForm();

this.Hide();

}

}

else MessageBox.Show("Введите данные.","Ошибка!");

}

bool auth(string login, string pass)

{

bool rezult = false;

for (int j = 0; j < this.dataSet.Пользователь.Rows.Count; j++)

{

if (this.dataSet.Пользователь.Rows[j][0].ToString() == login && this.dataSet.Пользователь.Rows[j][1].ToString() == pass)

{

user.setID(Convert.ToInt32(this.dataSet.Пользователь.Rows[j][3]));

user.setUserName(this.dataSet.Пользователь.Rows[j][2].ToString());

user.setGroup(this.dataSet.Пользователь.Rows[j][4].ToString());

rezult = true;

}

}

if(rezult == false) MessageBox.Show("Неправильный логин/пароль.", "Ошибка!");

return rezult;

}

После авторизации мы переходим к главной форме на которой происходит работа пользователя или администратора с приложением. На главной форме представлена справка о программе и пользователи, а также переход формы «пользователи», «Мероприятия», «Журнал мероприятий» и «Смена». Для отчетов была разработана кнопка в menuStrip «Файл» для открытия отчетов. Для запросов также была разработана кнопка в menuStrip «Функции». На рисунке 9 изображена главная форма.



Рисунок 9 – Главная форма.

Переход на формы реализован:

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

«название переменной».MdiParent = this;

«название переменной».Show();

Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, С++, так и на VB.Net, J# и др.

С одной стороны, Windows Forms рассматривается как замена более старой и сложной библиотеке MFC, изначально написанной на языке C++. С другой стороны, WF не предлагает парадигму, сравнимую с MVC. Для исправления этой ситуации и реализации данной функциональности в WF существуют сторонние библиотеки. Одной из наиболее используемых подобных библиотек является User Interface Process Application Block, выпущенная специальной группой Microsoft, занимающейся примерами и рекомендациями, для бесплатного скачивания. Эта библиотека также содержит исходный код и обучающие примеры для ускорения обучения.

## Разработка запросов

Вся работа с запросами работает с помощью ComboBox. Элемент ComboBox образует выпадающий список и совмещает функциональность компонентов ListBox и TextBox. Для хранения элементов списка в ComboBox также предназначено свойство Items.

В данном приложение есть 13 запросов:

* В форме «Пользователи»:
  + Запрос на смену пароля;
  + Запрос на добавления пользователя.

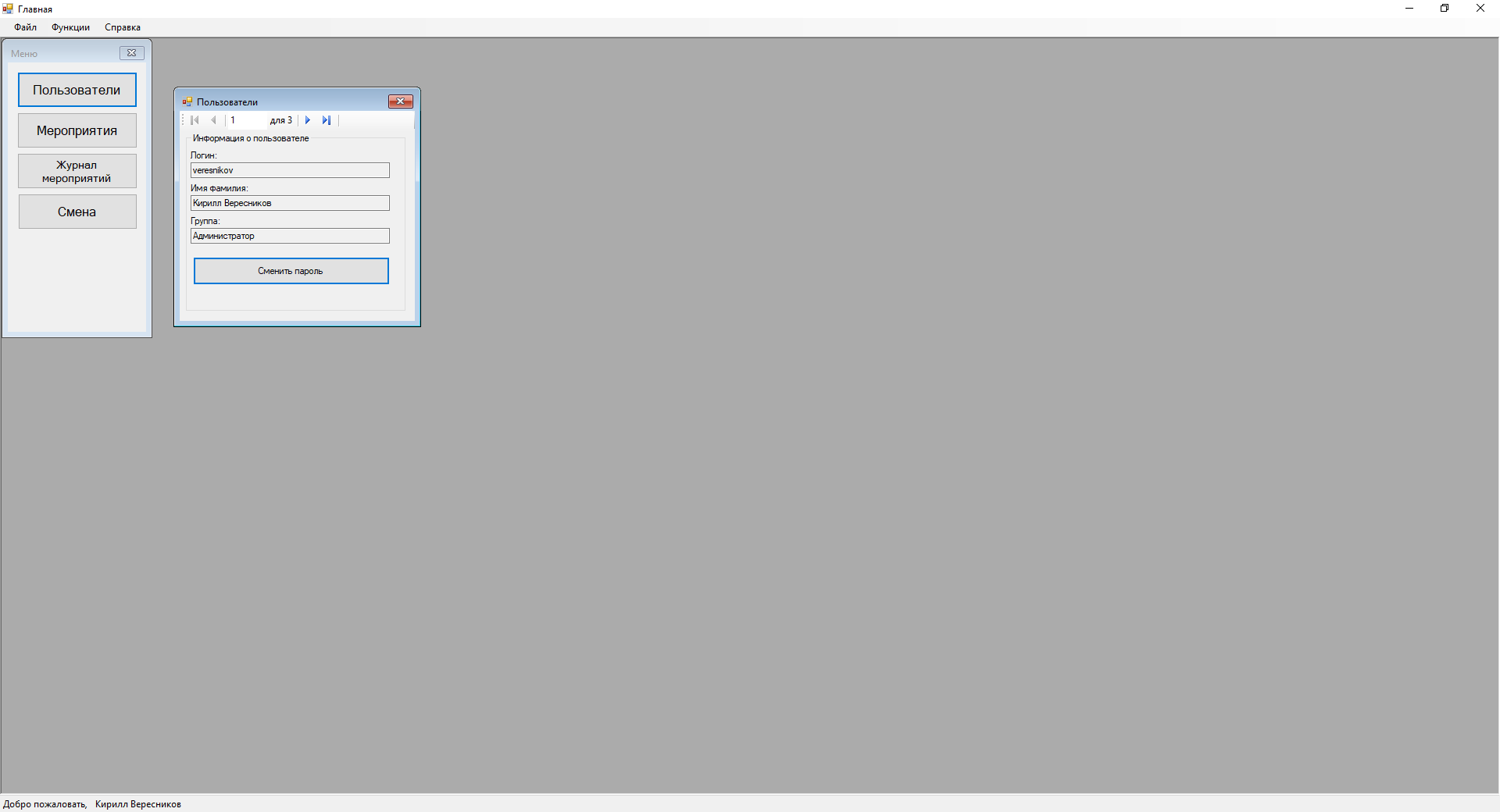


Рисунок 10 Форма пользователя.

* В форме «Мероприятия»:
  + Запросы на подстановку значений;
  + Запрос на добавления мероприятия;
  + Запрос на удаление мероприятия.

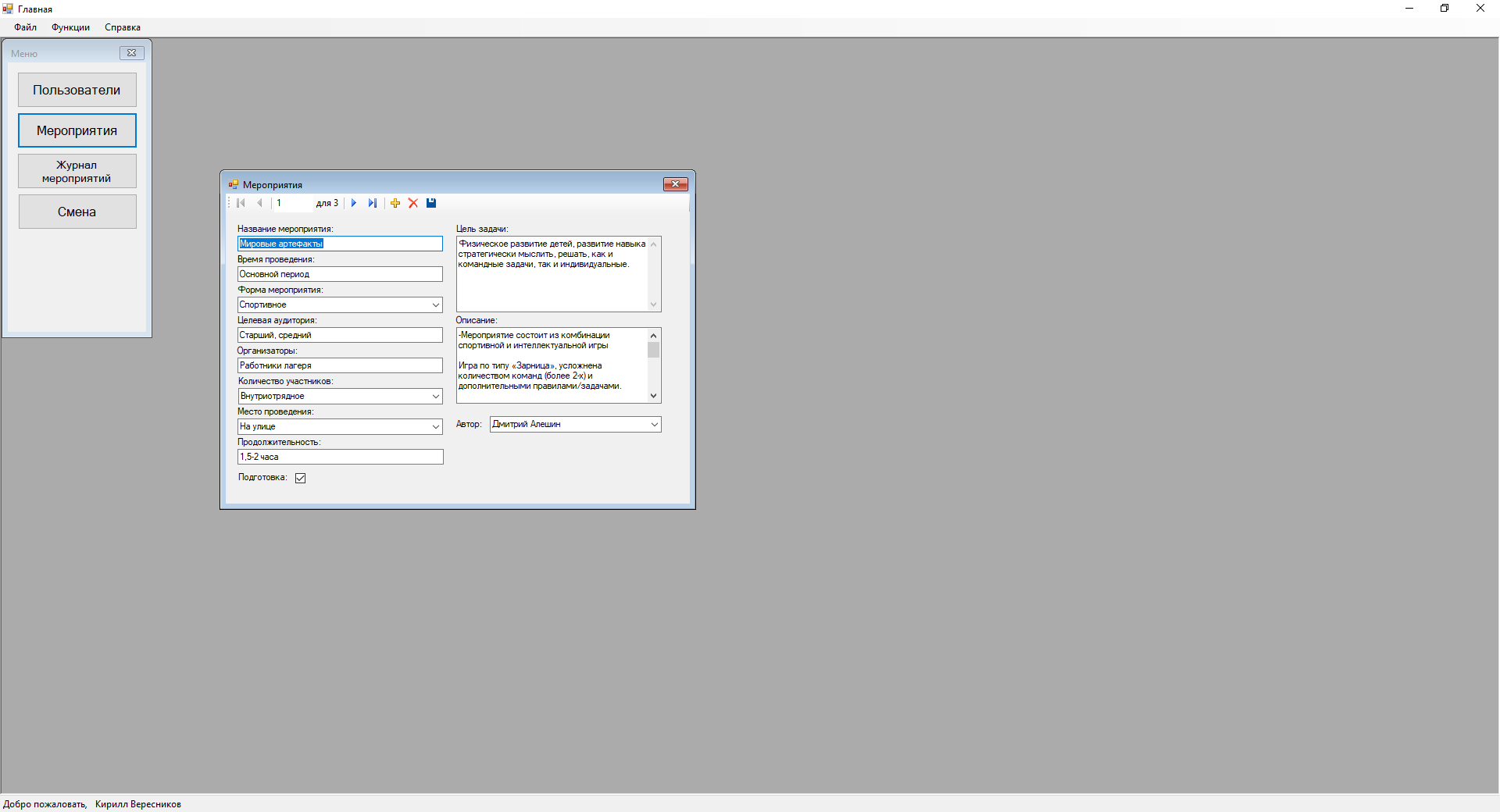


Рисунок 11 – Форма мероприятий.

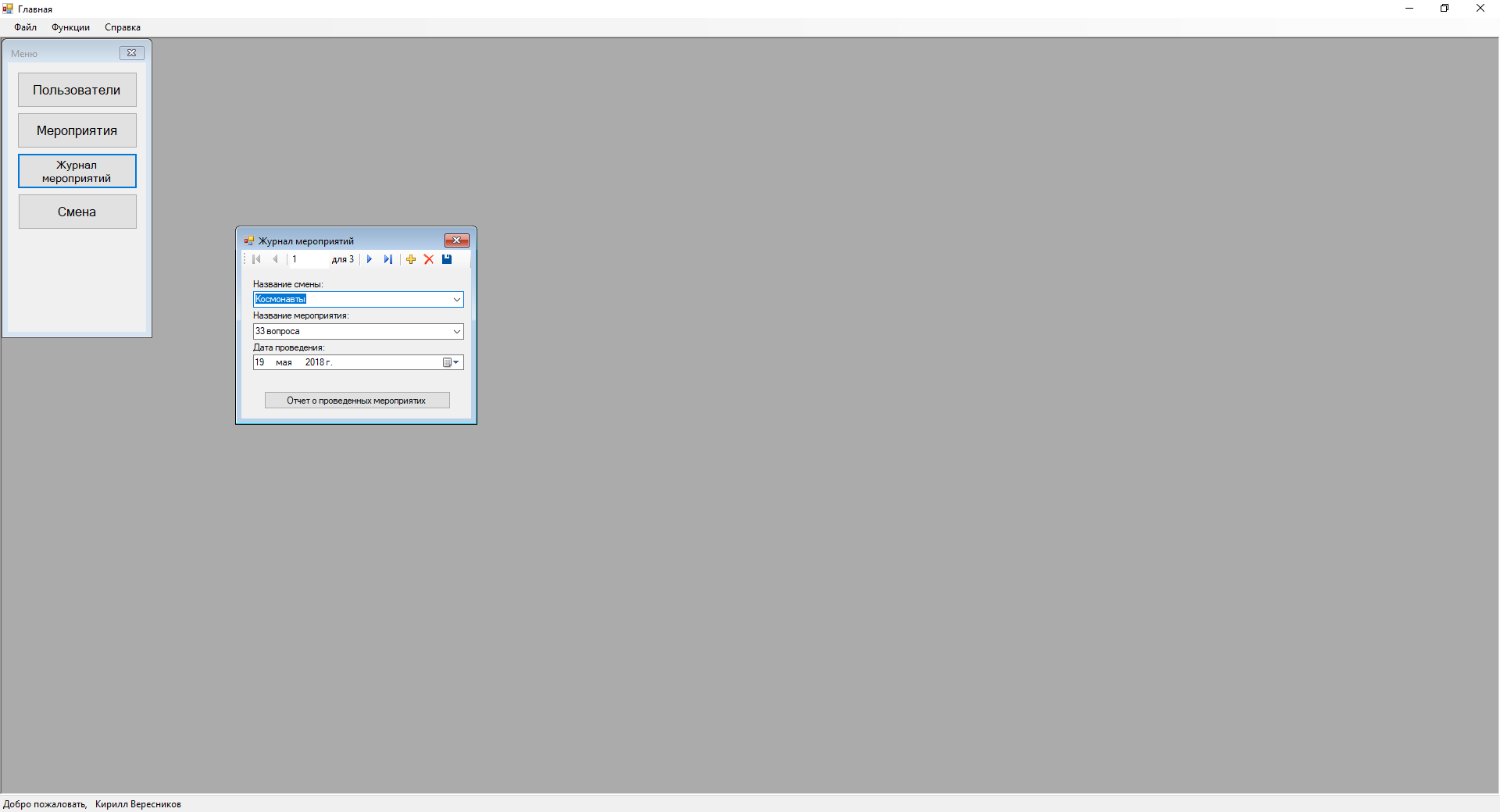
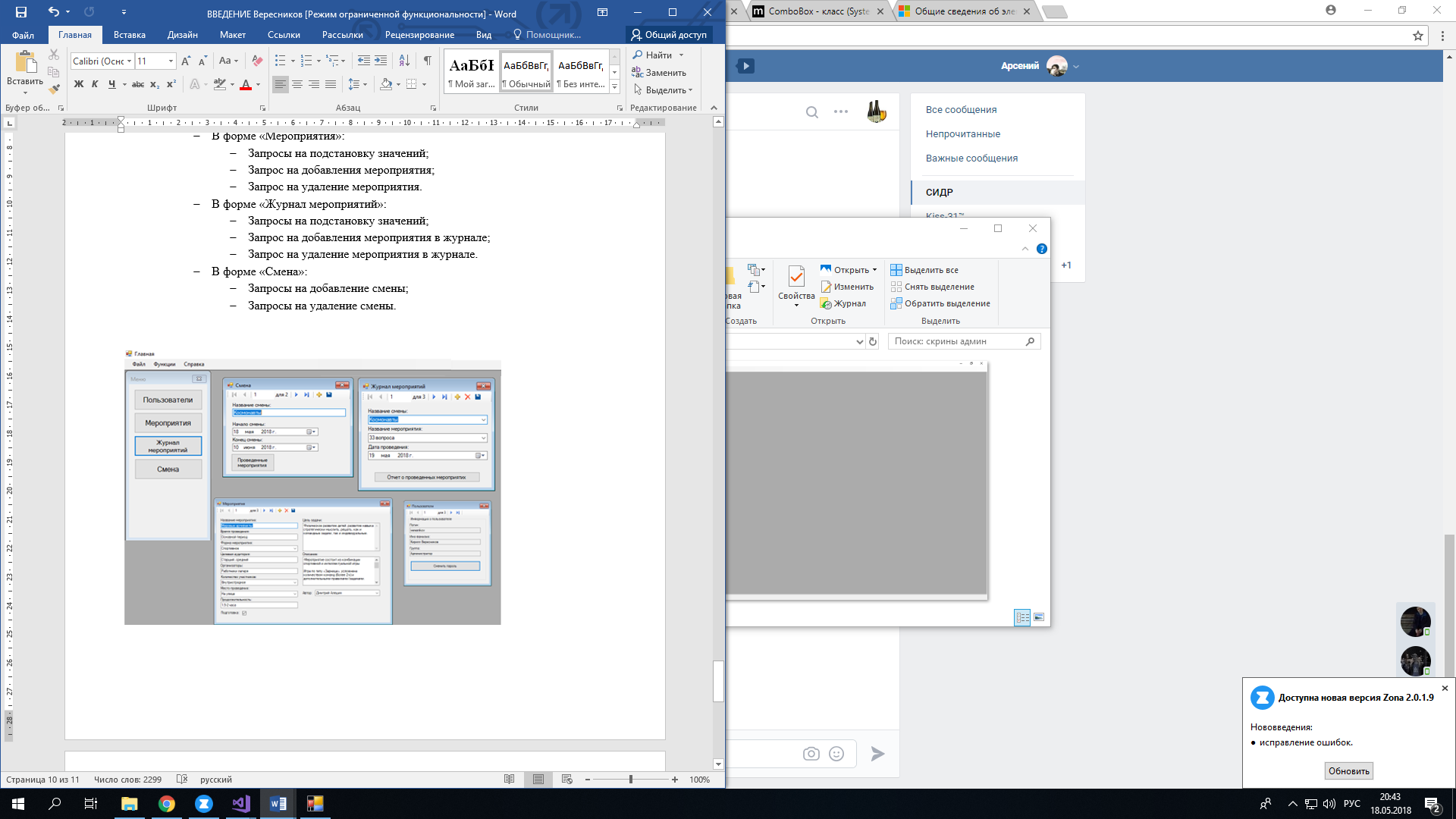
* В форме «Журнал мероприятий»:
  + Запросы на подстановку значений;
  + Запрос на добавления мероприятия в журнале;
  + Запрос на удаление мероприятия в журнале.

Рисунок 12 – Форма журнал мероприятий.

* В форме «Смена»:
  + Запросы на добавление смены;
  + Запросы на удаление смены.

Рисунок 13 –форма смена.

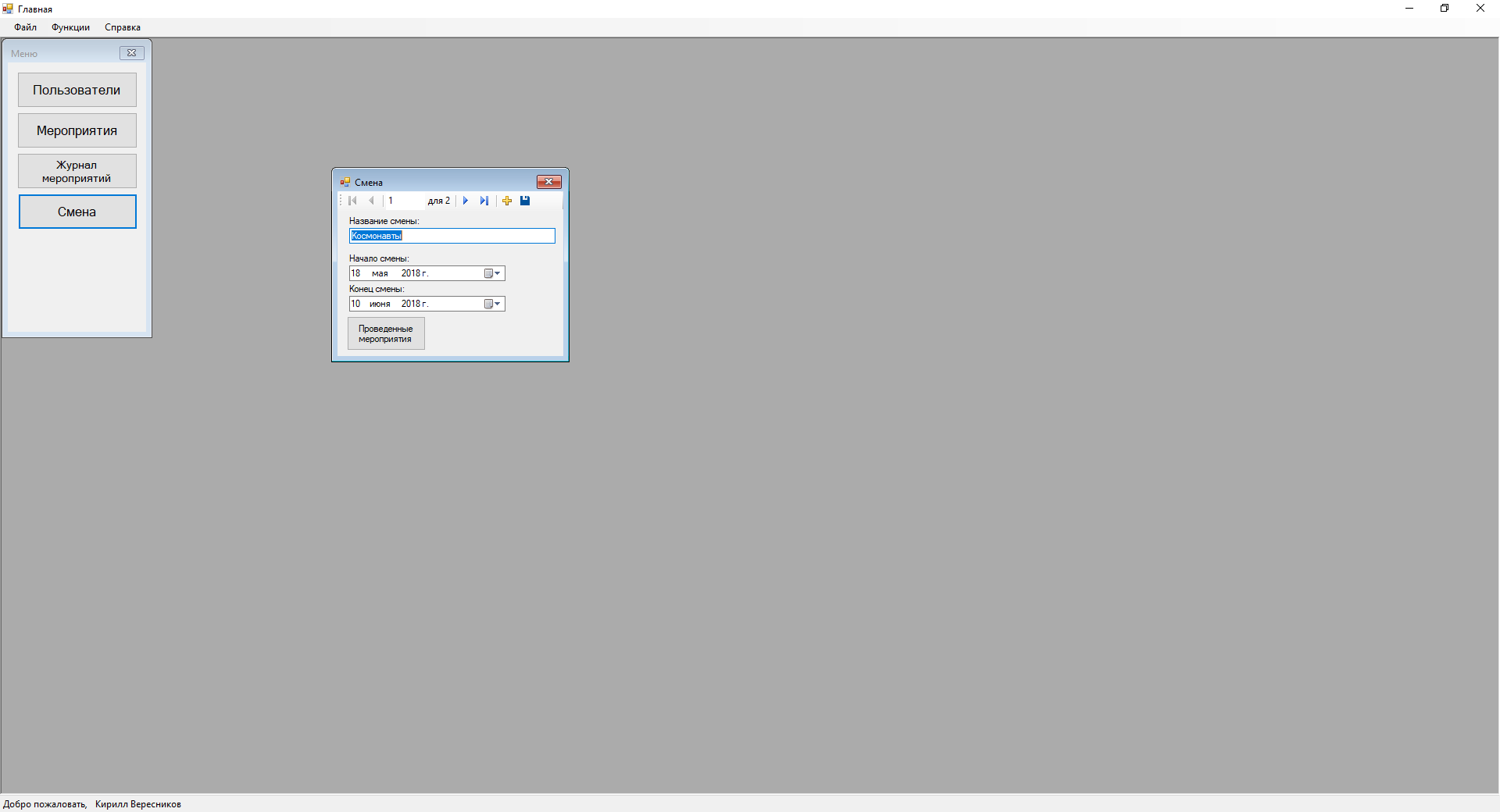


Рисунок 14 – Запросы

## Разработка отчетов

# Для разработки отчетов был использован модуль ReportViver для вывода данных из таблицы. Таким образом, были сформированы отчеты: по учету, о рейтинге учеников и о старших учениках. Microsoft Visual Studio 2017 поддерживает функции создания отчетов и элементов управления ReportViewer, которые позволяют добавлять полнофункциональные отчеты в пользовательские приложения. Отчеты могут содержать табличные, агрегатные и многомерные данные. Предоставляемые элементы управления ReportViewer позволяет обрабатывать и отображать отчеты в приложениях.

В данном приложении имеются 3 отчета:

* В форме «Отчет о проведенных мероприятиях»:
  + Отчет о проведенных мероприятиях.

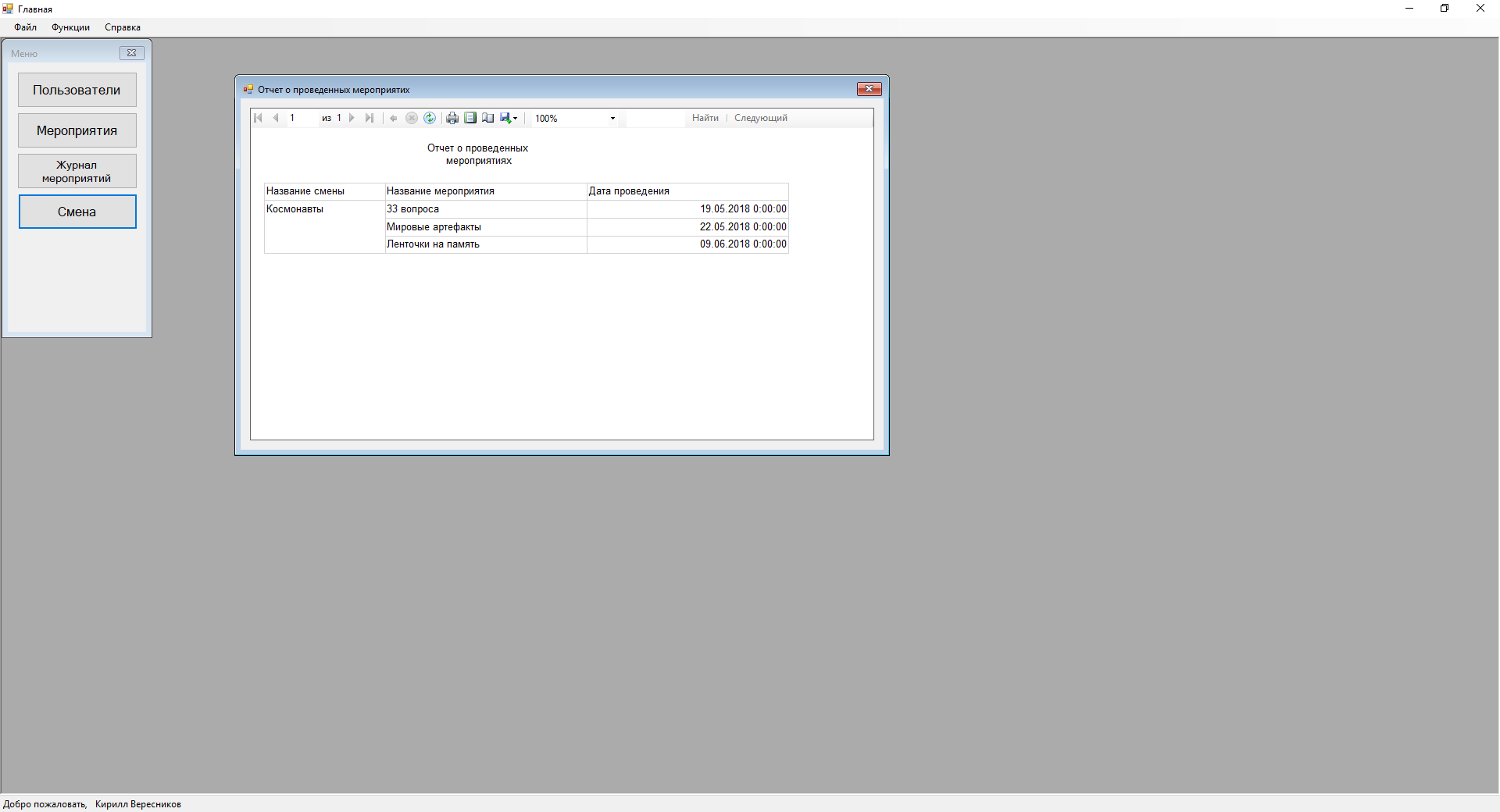
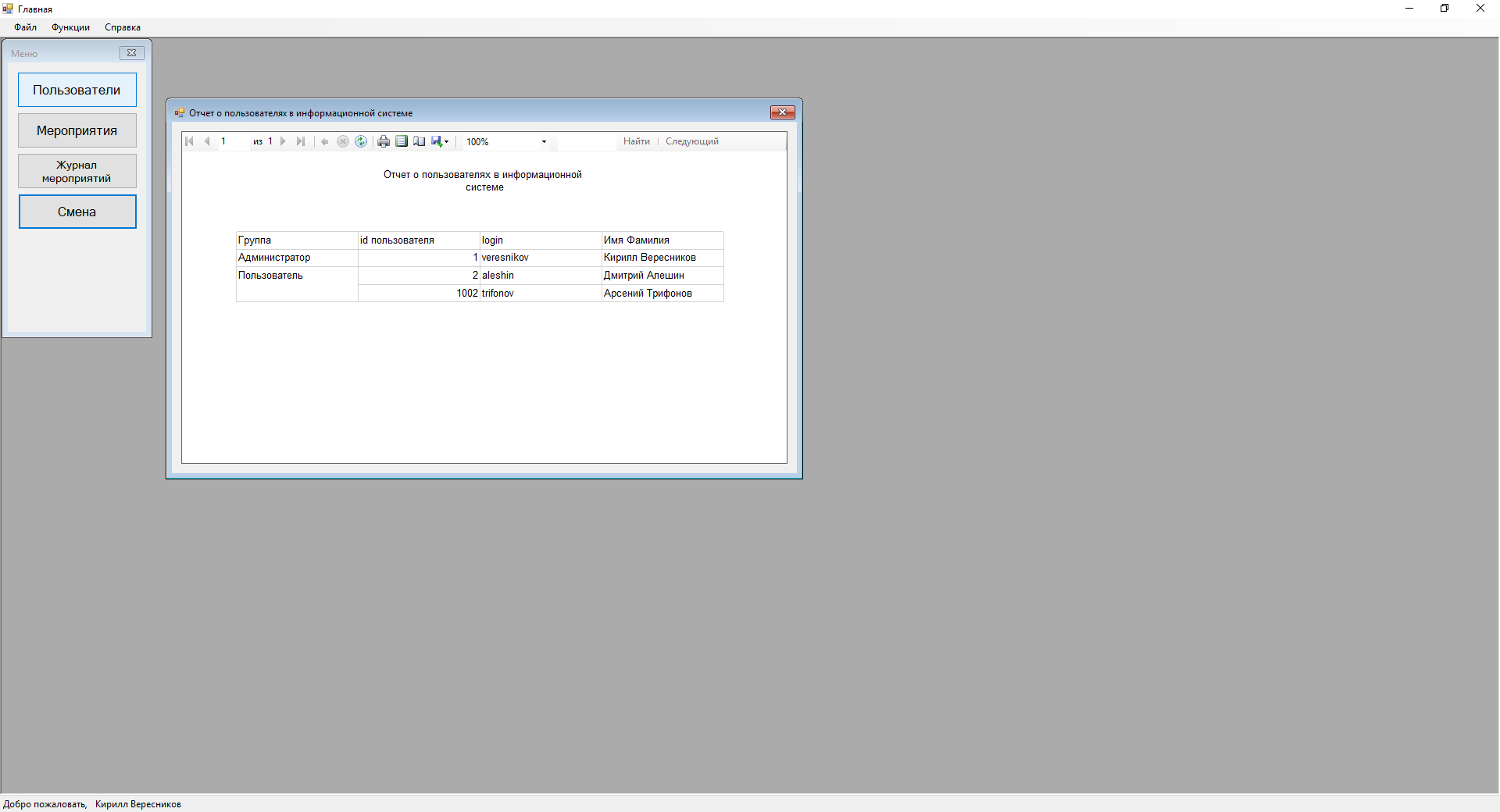


Рисунок 15 - отчет о проведенных мероприятиях

* В форме «Отчет о пользователях в информационной системе»:
  + Отчет о пользователях в информационной системе.

Рисунок 16 - Отчет о пользователях в информационной системе.



* В форме «Отчет о созданных пользователем мероприятий»:
  + Отчет о созданных пользователем мероприятий.

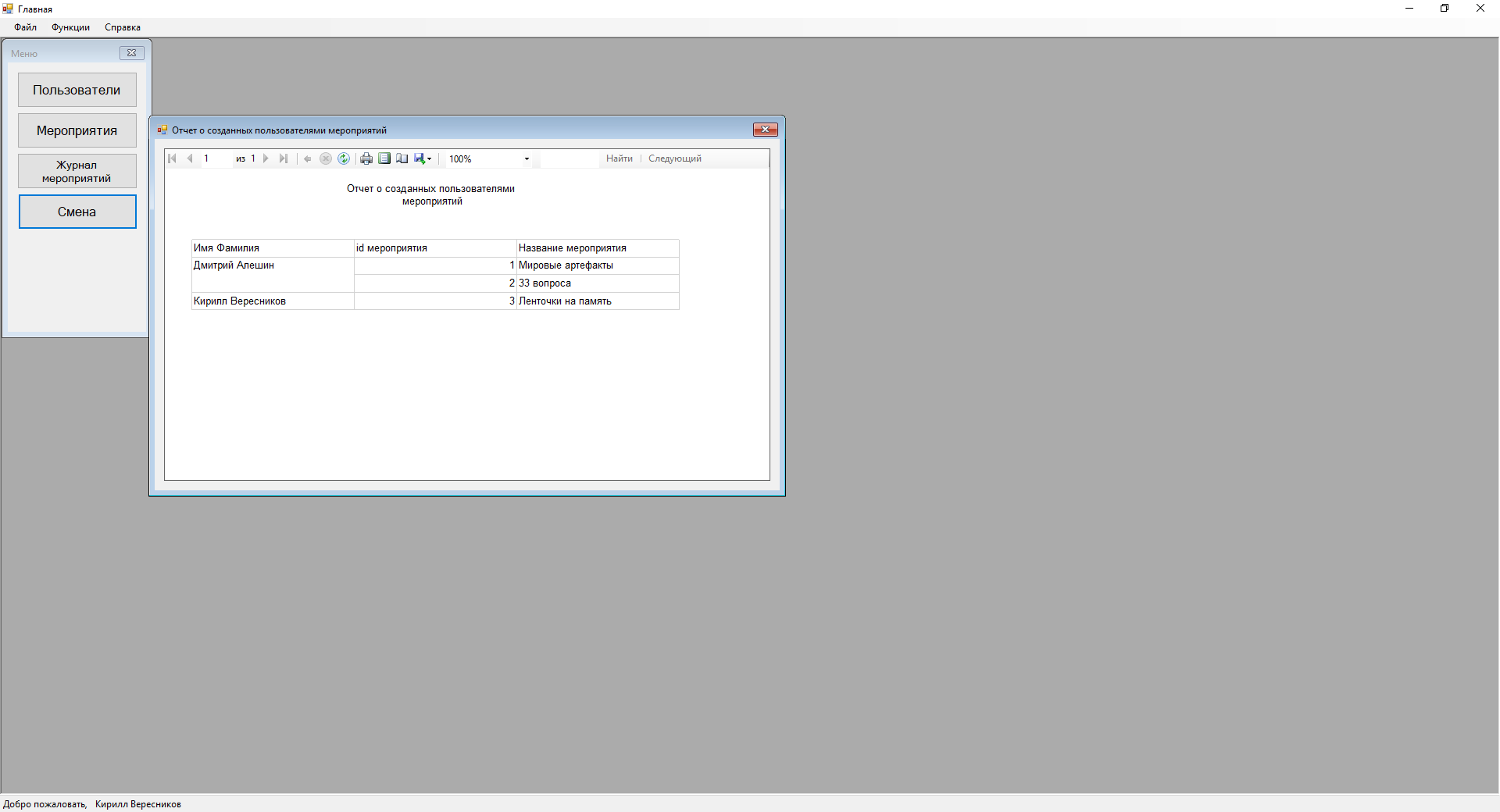


Рисунок 17 - Отчет о созданных пользователем мероприятий.

## Меню и справка

На главной форме находится элемент MenuStrip на котором расположена кнопка «справка «О программе» вызывающие Форму в котором будет описана справка для приложения и сведения об авторе. Их расположения представлены на рисунке 18 – О программе.

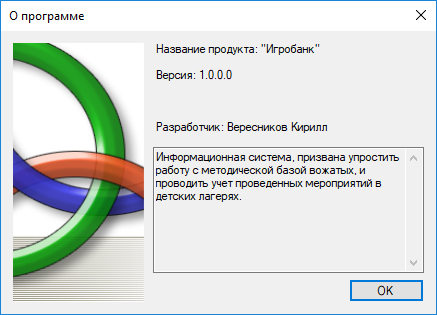


Рисунок 18 – О программе.

Главное меню было представлено пункте 2.2.2. Реализация пользовательских форм и показана на рисунке 9 – главная форма.

# ЗАКЛЮЧЕНИЕ

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

Разработанное приложение может быть использовано в детских оздоровительных лагерях.

Целью данной информационной системы является своевременное обеспечение надлежащих людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области, при этом результатом функционирования информационных систем является информационная продукция - документы, информационные массивы, базы данных и информационные услуги, и снижение рутиной работы с бумагами во время подбора мероприятий.

Во время выполнения курсового проекта были использованы следующие программные продукты:

* Visual Studio 2017
* MS SQL Server
* SQL Server Management Studio
* BPwin
* ERwin

Приложение может:

* формировать отчеты;
* осуществлять учет мероприятий;
* добавлять новые мероприятия в базу;

Выполняя данный курсовой проект, была более углубленно изучена работа с Visual Studio 2017 с MS SQL Server.

# Список литературы

1. Гагарина Л.Г. Разработка и эксплуатация АИС. – М., Форум, 2007
2. Голицына, О.Л. и др. Базы данных; Форум; Инфра-М, 2013
3. Емельянова Н.З. Основы построения автоматизированных информационных систем. – М., Форум, 2007
4. Олифер В.Г. Компьютерные сети. Принципы, технологии. - СПб., Питер, 2009
5. Сеннов А. Access 2007/ Учебный курс. – СПб., Питер, 2007
6. Баженова И. Ю. SQL и процедурно-ориентированные языки. - Национальный Открытый Университет «ИНТУИТ», 2016 г.
7. Гвоздева В.А. Основы построения автоматизированных информационных систем. – М., Форум, 2009
8. Дунаев В..В. Базы данных Язык SQL. – СПб., БХВ - Петербург, 2006
9. Кушнир А.Н. Microsoft Office. Access 2007
10. Максимов Н.В., Попов И.И. Компьютерные сети.- М, ФОРУМ: ИНФРА-М, 2010
11. Малыхина М.П. Базы данных: основы, проектирования, использование. – СПб., БХВ-Петербург, 2006
12. Мельников В.П. Информационная безопасность. – М., Академия, 2009
13. Лазицкас Е. А., Загумённикова И. Н., Гилевский П. Г. Базы данных и системы управления базами данных: учебное пособие. – РИПО, 2016 г.
14. Рудикова Л.В. Базы данных. Разработка приложений. – СПб., БХВ - Петербург, 2006
15. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5 - 6-е издание, 2012
16. Шумаков П. В. ADO.NET и создание приложений баз данных в среде Microsoft Visual Studio.NET: руководство разработчика с примерами на C#. - Диалог-МИФИ 2003 г.