СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#_Toc163493437)

[1 ОБЩАЯ ЧАСТЬ 3](#_Toc163493438)

[1.1 Информационная система предприятия 3](#_Toc163493439)

[1.2 Описание предметной области 4](#_Toc163493440)

[2 ПРОЕКТИРОВАНИЕ 5](#_Toc163493441)

[2.1 Проектирование диаграмм UML 5](#_Toc163493442)

[2.2 Проектирование диаграмм потоков данных 5](#_Toc163493443)

[2.3 Разработка модульной структуры системы 7](#_Toc163493444)

[3 РЕАЛИЗАЦИЯ 8](#_Toc163493445)

[3.1 Настройка и подключение система контроля версий 8](#_Toc163493446)

[3.2 Разработка программного модуля 9](#_Toc163493447)

[3.2.1 Разработка объектов базы данных 9](#_Toc163493450)

[3.2.2 Реализация CRUD 11](#_Toc163493451)

[3.2.3 Фильтрация и поиск данных 16](#_Toc163493452)

[4 Тестирование 17](#_Toc163493453)

[4.1 Smoke-тестирование 17](#_Toc163493454)

[4.2 Тест-кейсы 17](#_Toc163493455)

[5 Вывод 17](#_Toc163493456)

[6 Приложение 17](#_Toc163493457)

# ВВЕДЕНИЕ

Данная работа посвящена управлению и автоматизации баз данных для школы. Актуальность темы заключается в том, что база данных для школы является крайне важным инструментом для эффективной работы и управления. Она помогает автоматизировать процессы, упростить учет и отслеживание учеников, оптимизировать управление уроками и различными секциями. Кроме того, она позволяет улучшить качество обслуживания клиентов и повысить уровень их удовлетворенности.

Целью производственной практики является создание и администрирование базы данных по учету информации об организации школы. Использование базы данных позволило скомпоновать необходимые данные в нескольких каталогах. В данном программном средстве можно выделить следующие главные части: получение сведений об учителях, информации об учениках и их успеваемости, а также уроках и их недельное расписание.

Для выполнения цели необходимо выполнить следующие задачи:

* Настройка и проектирование систем управления баз данных;
* Реализация клиентского приложения;
* Использование и реализация объектов баз данных;
* Освоение администрирования базы данных.

Объектом исследования в производственной практике является: школа.

Предметом исследования является: база данных школы.

# ОБЩАЯ ЧАСТЬ

## Информационная система предприятия

Полное наименование образовательной организации: Муниципальное бюджетное общеобразовательное учреждение "Райковская средняя общеобразовательная школа имени Николая Ильича Носова".

Сокращенное наименование: МБОУ "Райковская СОШ им. Н.И. Носова".

Общее описание предприятия: МБОУ “Райков. СОШ” 1977 года постройки, функциональна с 1978 года, муниципальная собственность Усть-Абаканский района, имеет СПДО детский сад “сказка”. Образовательная деятельность: лицензия № 2144 от 29.08.2016 года.

Структура предприятия: Управление школой осуществляется в соответствии с законодательством Российской Федерации, Уставом образовательного учреждения и строится на принципах единоначалия и коллегиальности.

Единоличным  исполнительным органом МБОУ «Райковская СОШ имени Н.И. Носова» является директор, который осуществляет текущее руководство  деятельностью образовательного учреждения.

Органами коллегиального управления являются:

* Общее собрание работников;
* Педагогический совет.

Общее собрание работников МБОУ «Райковская СОШ имени Н.И. Носова» является постоянно действующим органом самоуправления образовательного учреждения. Членами Общего собрания являются все работники ОО. Общее собрание действует на основании Закона Российской Федерации от 27.12.2012 № 273 – ФЗ «Об образовании в Российской Федерации», Трудового кодекса РФ, нормативных правовых документов в области труда в образовании, Устава ОО, Положения об общем собрании работников. Решения Общего собрания являются рекомендательными для коллектива ОО. Решения Общего собрания, утвержденные приказом по ОО или распоряжение вышестоящей организации, являются обязательными для исполнения.

Педагогический совет является постоянно действующим органом школы для рассмотрения основных вопросов учебно-воспитательной работы. Главными задачами педагогического совета являются: объединение усилий педагогического коллектива школы на повышение уровня образовательной деятельности, внедрение в практику достижений науки и передового педагогического опыта.

Правила внутреннего распорядка: Учебные занятия начинаются в 8:30,а заканчиваются в 15:15. Также организация имеет дополнительные занятия, кружки, которые длятся до 20:00.

## Описание предметной области

Школа - учебное заведение для получения общего образования. В широком смысле, данное слово способно применяться в качестве названия для остальных образовательных учреждений.

Время в школе разделяется на следующие этапы: внеурочная деятельность, перемена, урок. Урок и перемена, как правило, сменяют друг друга несколько раз. Завершается же учебный день обычно внеурочной деятельностью.

Урок  - занятие школьников по предметам.

Перемена  - отдых между уроками.

Внеурочная деятельность – это такая система дополнительного образования, при которой ученик располагается в школе после уроков.

Ежедневно в школе проводится от четырех до восьми уроков, а также различные внеурочные деятельности. Именно поэтому организации необходимо вести постоянно расписание уроков для сведения в единую взаимосвязанную систему учащихся, учителей, уроков и назначенных для проведения занятий мест — классов.

Школа, как всякая организация, существует не изолированно, а в определенной социальной среде. Она является сложной целостной системой, активно взаимодействующей с внешней средой.

Именно поэтому важными аспектами в данной предметной области являются быстрое технологическое развитие, быстрое появление новых ресурсов, а также сильное развитие информационной структуры. Поэтому различные школы должны постоянно следить за новинками, предоставлять качественное образование ученикам, быть готовыми к постоянному развитию технологий и иметь высококвалифицированных специалистов для работы.

# ПРОЕКТИРОВАНИЕ

## Проектирование диаграмм UML

Диаграмма в языке моделирования UML — наглядное представление некоей совокупности элементов модели системы в виде графа, на котором отношения связывают сущности. В своём графическом виде различные виды диаграмм UML применяются для визуализации разных аспектов устройства или поведения моделируемой системы.

На рисунке 1 изображена диаграмма вариантов использования, на которой показаны два актёра «Пользователь», «Администратор», а также из варианты использования.

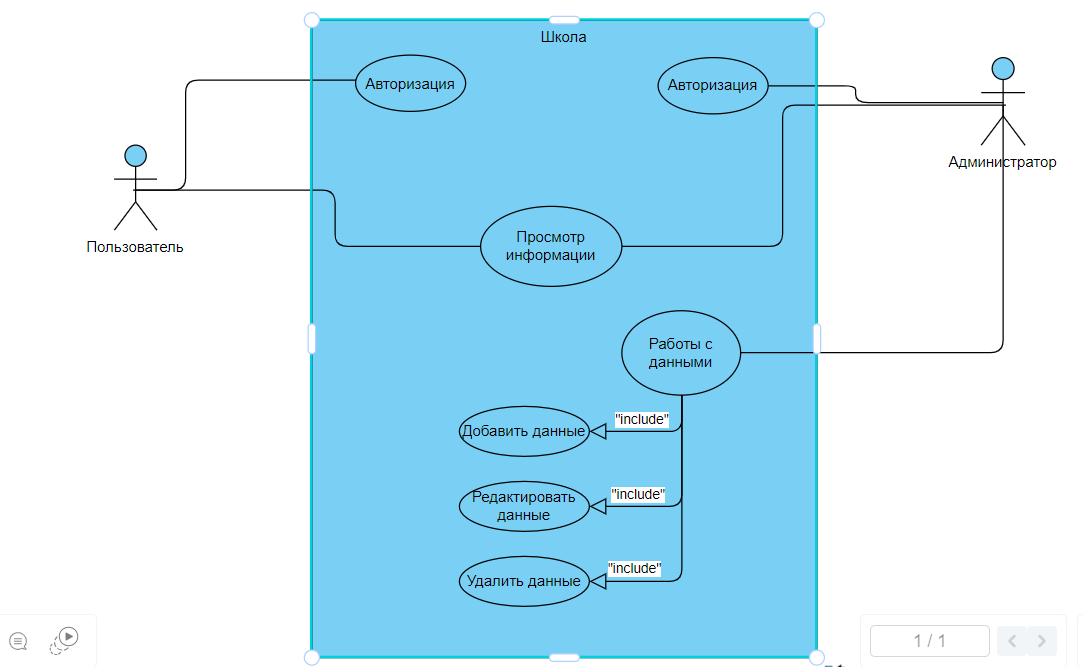


Рисунок 1 – Диаграмма вариантов использования

## Проектирование диаграмм потоков данных

Диаграмма потоков данных — один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения [UML](https://ru.wikipedia.org/wiki/UML).

DFD 0-го уровня также называется контекстной схемой. Это простейший способ изображения анализируемых или моделируемых систем и процессов. Такие схемы показывают общую картину и представляют систему в виде единого процесса, наделенного связями с внешними сущностями. Схемы 0-го уровня будут понятны широкой аудитории, включая участников проекта, бизнес-аналитиков и разработчиков.

DFD 1-го уровня дает более детальное представление об элементах контекстной схемы. Разбив обобщенный процесс контекстной схемы на подпроцессы, получится выделить основные функции системы.

На рисунке 2 изображена диаграмма потоков данных IDF-0.

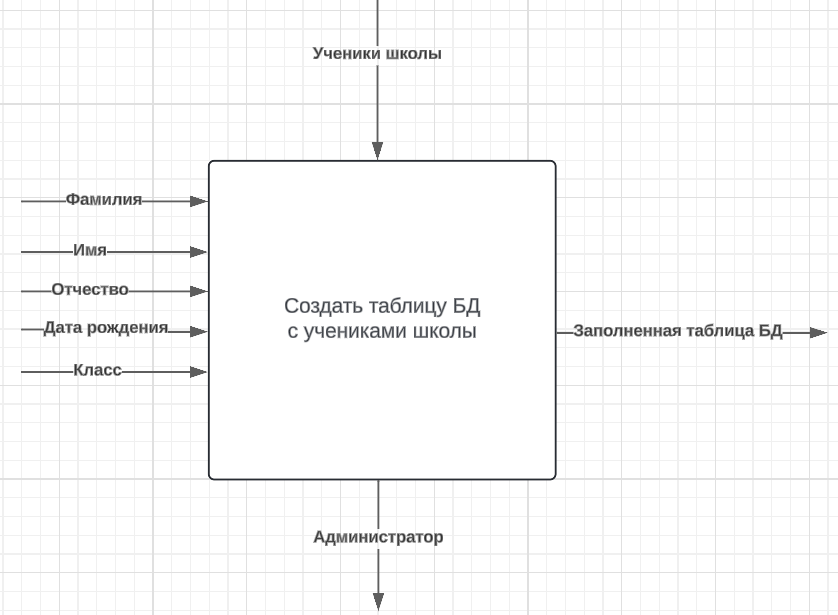


Рисунок 2 – Диаграмма потоков данных IDF-0

На рисунке 3 изображена диаграмма потоков данных IDF-1.

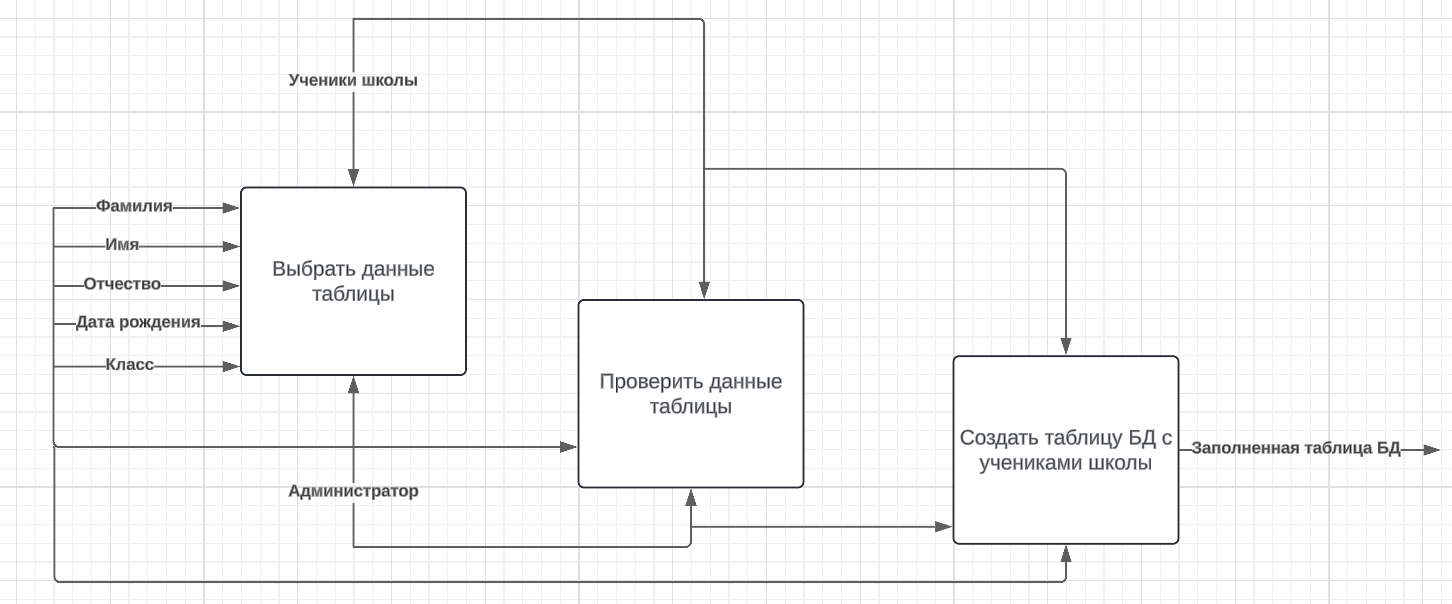


Рисунок 3 – Диаграмма потоков данных IDF-1

## Разработка модульной структуры системы

Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Данная диаграмма представляет две сущности «Администратор» и «Пользователь» и то, как эти сущности будут взаимодействовать с программным модулем. В данном случае «Администратор» может полностью работать с программным модулем, а именно просматривать, добавлять, удалять и редактировать данные БД. В свою очередь как «Пользователь» может только просматривать данные БД.

На рисунке 4 изображена диаграмма классов для «Администратора» и «Пользователя».

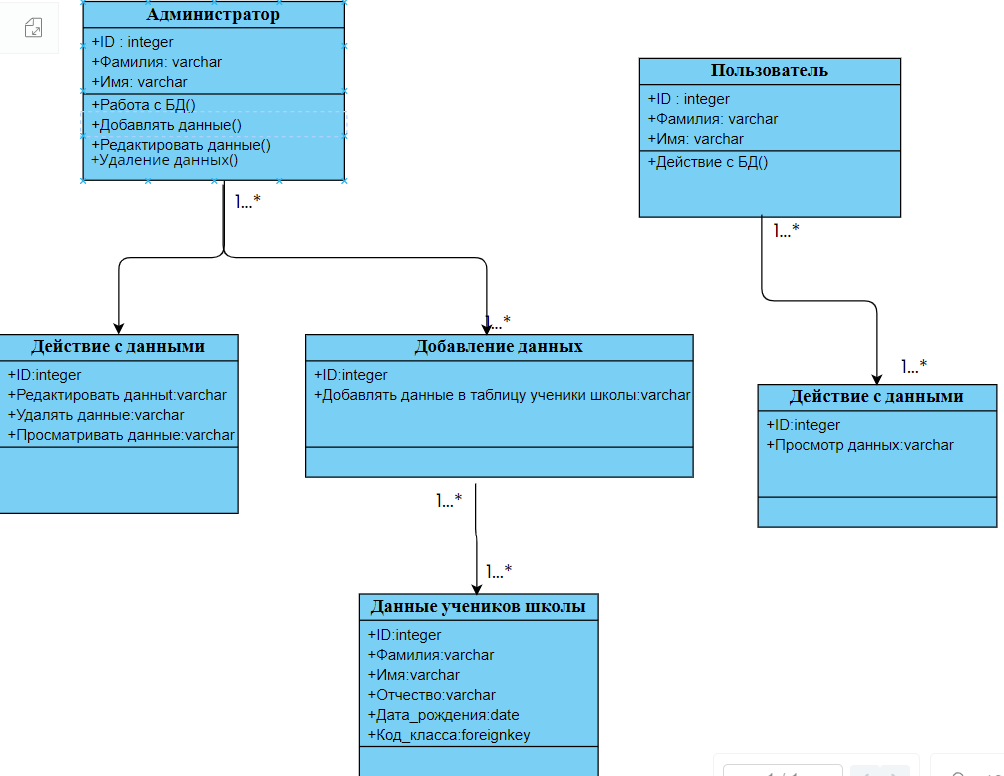


Рисунок 4 – Диаграмма классов

# РЕАЛИЗАЦИЯ

## Настройка и подключение система контроля версий

Для того чтобы выгрузить проект для начала необходимо установить приложение GitHub Desktop, с его помощью и будет выкладываться новые версии приложения. После установки необходимо авторизоваться на GitHub. Уже, после чего создаем новый репозиторий.

На рисунке 5 изображено создание нового репозитория.

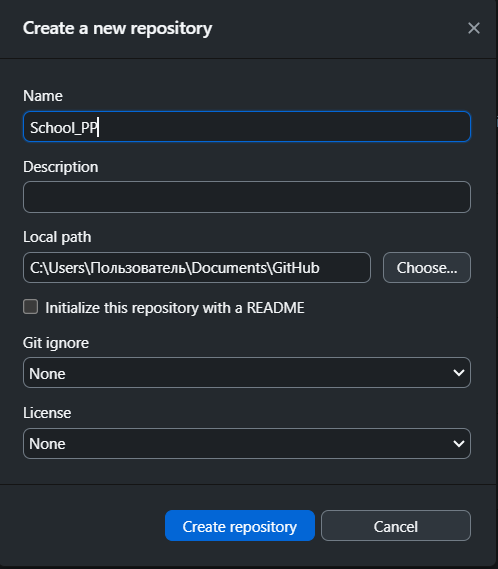


Рисунок 5 – Создание нового репозитория

После создания репозитория папку с разрабатываемым приложением необходимо перенести в только что созданный репозиторий, при этом комментируя изменения, внесённые в репозиторий, вызванные добавлением папки с приложением.

На рисунке 6 изображено комментирование изменений.

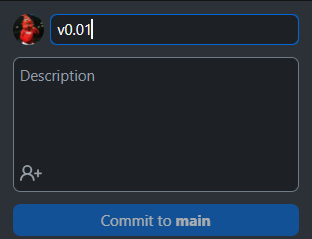


Рисунок 6 – Комментирование изменений

После внесения изменений публикуем репозиторий.

На рисунке 7 изображена публикация репозитория.

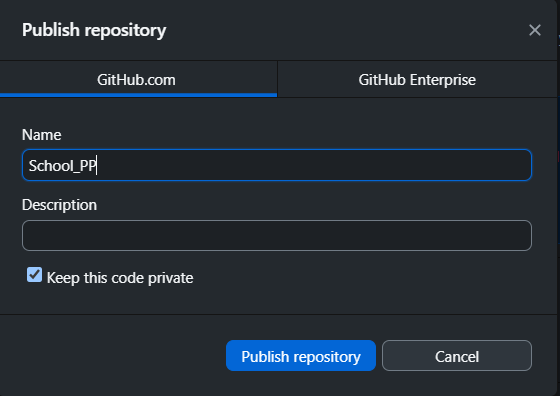


Рисунок 7 – Публикация репозитория

На рисунке 8 изображен итог публикации.

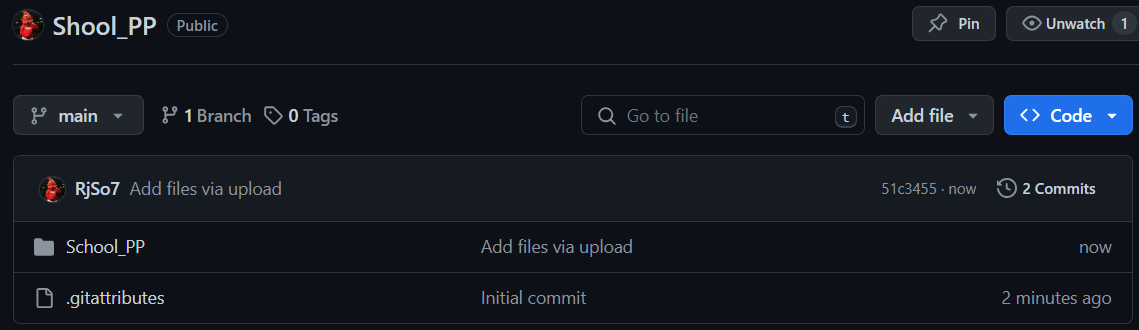


Рисунок 8 – Итог публикации

## Разработка программного модуля



### Разработка объектов базы данных

Создание таблицы происходит в СУБД MySQL. В данной СУБД можно создавать базы данных, таблицы к ним, а также администрировать их. Это можно сделать как при помощи ввода SQL-запросов, так и при помощи использования графического интерфейса, который называется MySQL Workbench.

Создание БД происходит на специальной вкладке «Schemas», где хранятся все БД. В данном меню мы можем настроить имя базы данных, а также ее кодировку. По окончанию настраивания параметров, необходимо нажать «Применить», после чего база данных создастся и появится в списке существующих баз.

На рисунке 9 изображено меню создания БД.

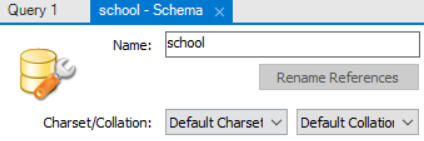


Рисунок 9 – Меню создания БД

Следующим этапом является создание таблиц данных, их заполнение, а также создание связей между ними. Создание связей между таблицами необходимо для обеспечения целостности данных, чтоб данные не терялись и не вызывали аномалий между собой.

Для начала создадим одну из основных таблиц БД «Кружки». Для этого необходимо развернуть список элементов базы данных, выбрать «Tables», далее в контекстном меню выбрать «Create Table», после чего откроется специальное меню создания таблицы данных. В этом меню необходимо задать название таблице, столбцам таблицы, указать тип принимаемых значений, а также установить первичный ключ, если в этом есть необходимость. Окончание создания таблицы сопровождается нажатием кнопки «Применить», где необходимо подтвердить действие, после чего таблица будет создана. В качестве примера приведем создание двух ключевых таблиц БД «кружки» и «ученики».

На рисунке 10 изображено создание таблицы «кружки».

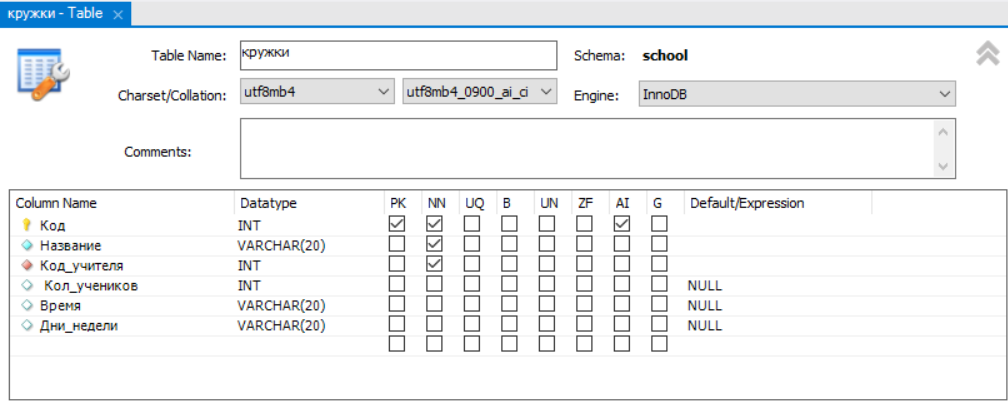


Рисунок 10 – Создание таблицы «кружки»

На рисунке 11 изображено заполненная таблица «кружки».



Рисунок 11 – Заполненная таблица «кружки»

На рисунке 12 изображено создание таблицы «ученики».

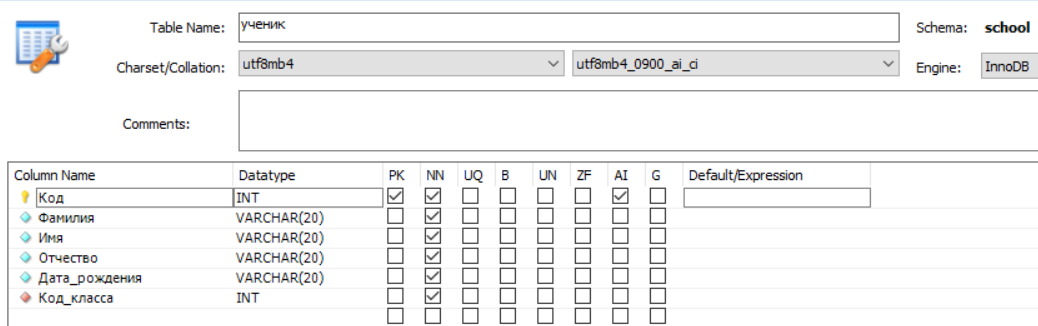


Рисунок 12 – Создание таблицы «ученики»

На рисунке 13 изображено заполненная таблица «ученики».

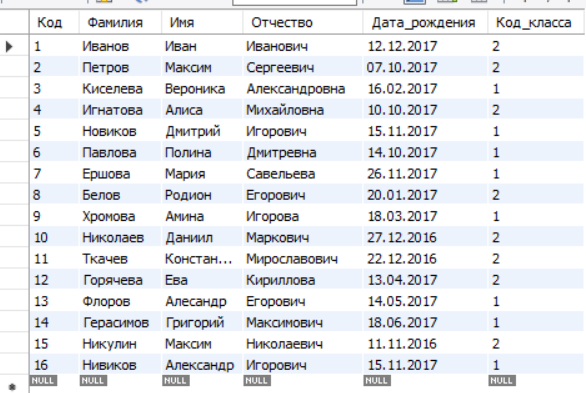


Рисунок 13 – Заполненная таблица «ученики»

Создание остальных таблиц БД были произведены аналогичным способом.

На рисунке 14 изображены все таблицы БД.

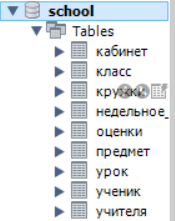


Рисунок 14 – Все созданные таблицы БД

### Реализация CRUD

При запуске приложения в первую очередь нас сопровождает окно авторизации, на которой необходимо ввести логин и пароль.

По нажатию кнопки выполняется проверка пользователей, либо вход будет выполнен с правами администратора, либо пользователя. Это сделано для того, чтобы предоставить или наоборот ограничить доступ к элементам управления формы.

На рисунке 15 изображено окно авторизации.

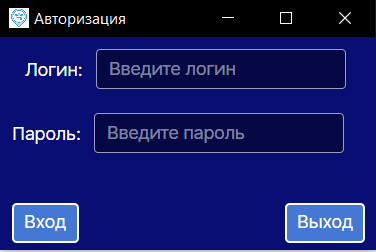


Рисунок 15 – Окно авторизации

После того как пользователь ввел логин и пароль, он переходит на главную форму приложения, при помощи которой можно просматривать таблицы базы данных, если вход был выполнен с правами пользователя или же взаимодействовать с функционалом приложения, но уже при помощи прав администратора.

На рисунке 16 изображено главное окно приложения.

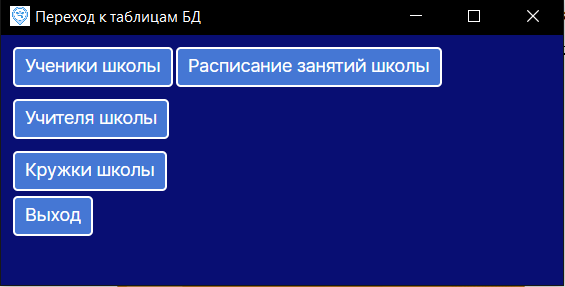


Рисунок 16 – Главное окно

После того как пользователь или администратор выберет необходимую ему таблицу БД, он перейдет на окно с отображением этой таблицы и предоставленным к ней функционалом.

На рисунке 17 изображено окно с отображением таблицы «ученики», если пользователь зашел от имени администратора.

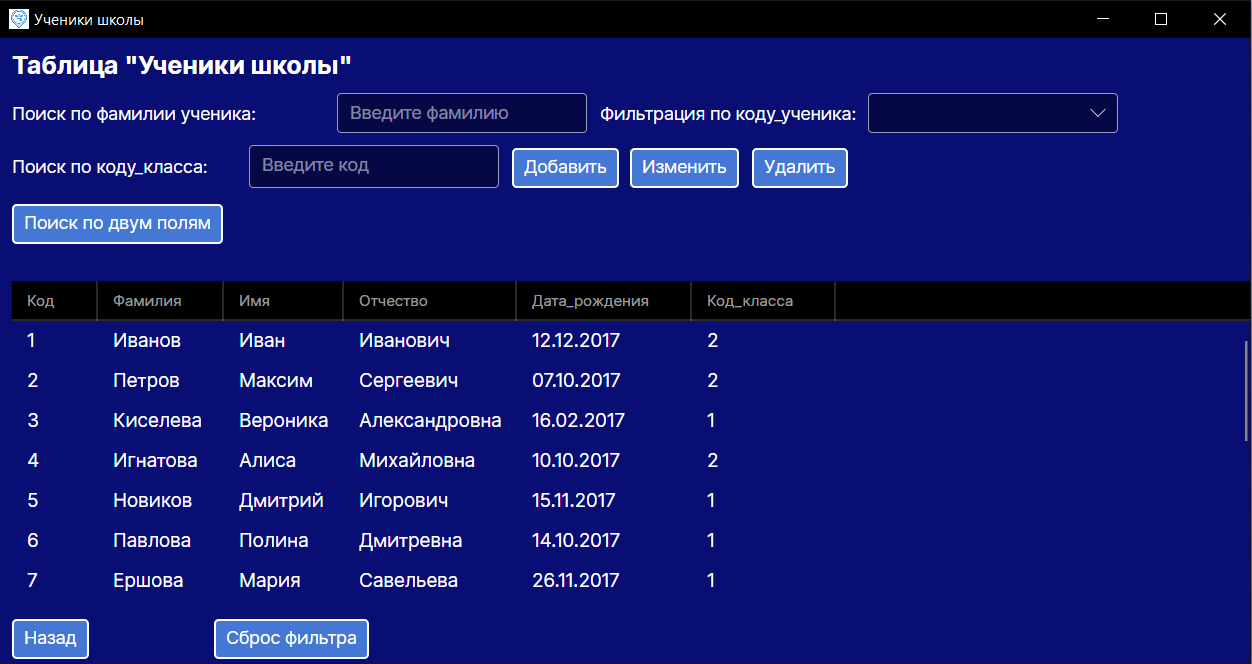


Рисунок 17 – Окно таблицы «ученики» с правами администратора

На рисунке 18 изображено окно с отображением таблицы «услуги», если пользователь зашел от имени обычного пользователя.

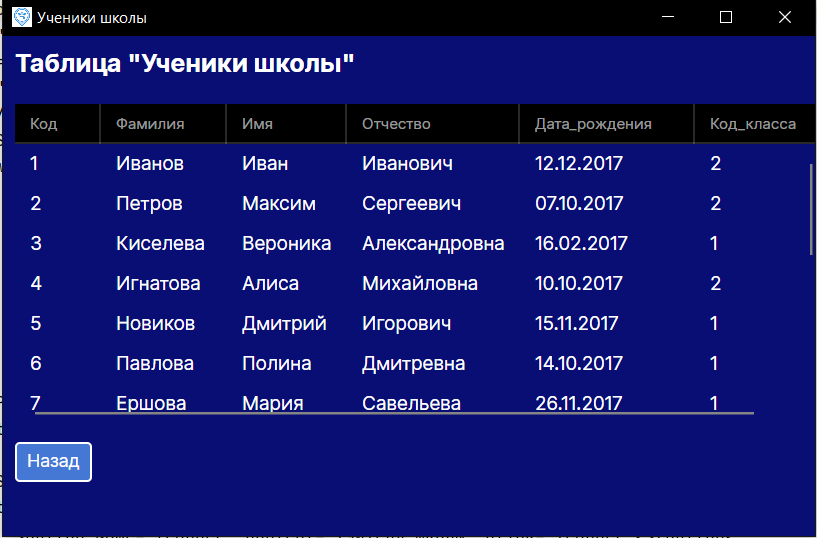


Рисунок 18 – Окно таблицы «услуги» с правами обычного пользователя

Отображение элементов на данном окне установлено таким образом, что некоторые из элементов скрываются или наоборот отображаются в зависимости от того, под какой учетной записью был выполнен вход. Это обеспечит более распределенное функционирование, а также гибкое управление приложением и базой данных. В данном случае обычный пользователь может только просматривать содержимое таблицы, а администратор использовать все возможные ему элементы управления для более быстрой и удобной работы с БД.

Вход с правами администратора позволяет добавлять, редактировать, удалять данные в таблице БД. При добавлении и редактировании данных открывается специальное окно, где необходимо внести данные или же изменить уже существующие старые данные.

На рисунке 19 изображен процесс добавления данных.

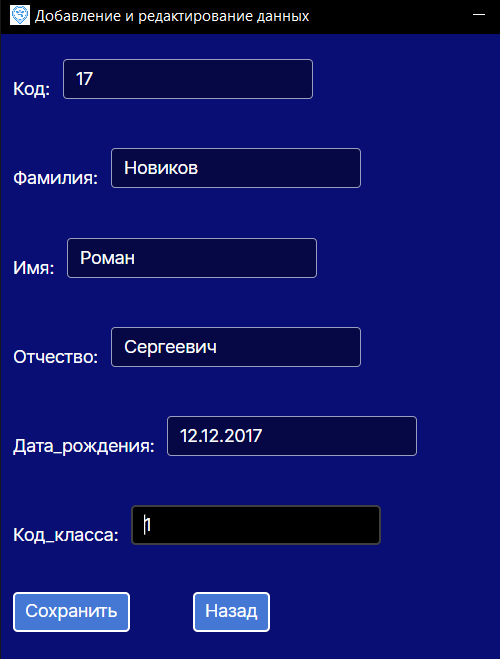


Рисунок 19 – Добавление данных

На рисунке 20 изображен итог добавления.

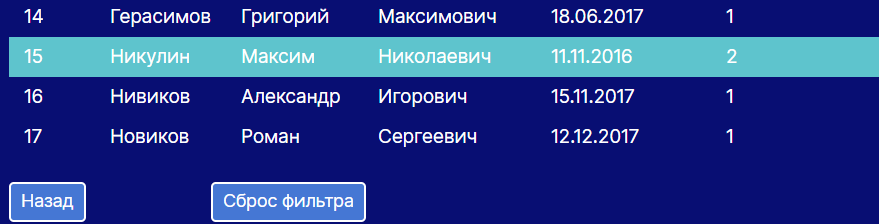


Рисунок 20 – Итог добавления

Для того чтобы начать редактировать запись, необходимо выбрать нужную запись и нажать на кнопку «Изменить». После нажатия кнопки, данные из полей будут перенесены в текстовые поля формы, где можно будет редактировать нужное поле.

На рисунке 21 изображен процесс редактирования данных.

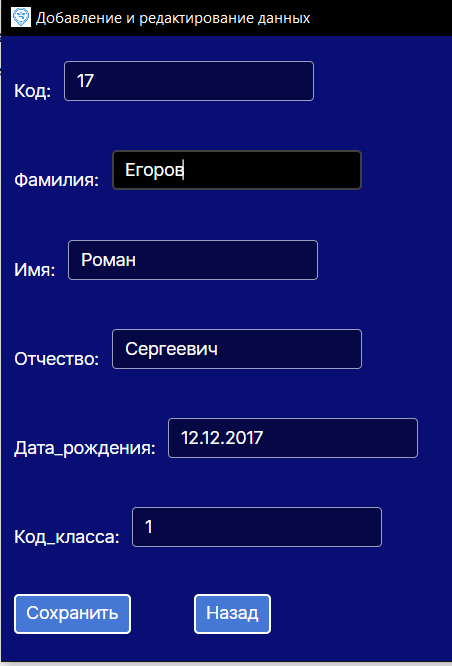


Рисунок 21 – Редактирование данных

На рисунке 22 изображен итог редактирования.

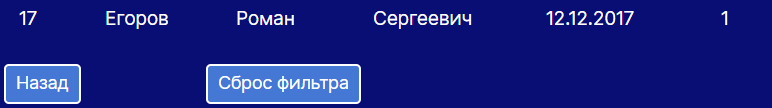


Рисунок 22 – Итог редактирования

Удаления работает аналогично редактированию, только вместо определенной записи необходимо выбрать нужный уникальный идентификатор таблицы и нажать на кнопку «Удалить». После чего запись удалится из таблицы и соответственно из БД.

На остальных таблицах добавление, удаление, редактирование происходит аналогичным способом.

На листинге 1 изображен код добавления и редактирования данных.

Листинг 1- код добавления и редактирования данных.

private void Save\_OnClick(object? sender, RoutedEventArgs e)  
{  
 var studentt = Studentt.FirstOrDefault(x => x.Код == \_currentStudent.Код);  
 if (studentt == null)  
 {  
 try  
 {  
 conn = new MySqlConnection(connStr);  
 conn.Open();  
 string add = "INSERT INTO ученик VALUES (" + Convert.ToInt32(Kod.Text)+ ", '" + Familia.Text + "', '" + Namee.Text + "', '" + Otch.Text +"','" + Date.Text +"', " + Convert.ToInt32(Klass.Text)+ ");";  
 MySqlCommand cmd = new MySqlCommand(add, conn);  
 cmd.ExecuteNonQuery();  
 conn.Close();  
 }  
 catch (Exception exception)  
 {  
 Console.WriteLine("Error" + exception);  
 }  
 }  
 else  
 {  
 try  
 {  
 conn = new MySqlConnection(connStr);  
 conn.Open();  
 string upd = "UPDATE ученик SET Фамилия = '" + Familia.Text + "', Имя = '" + Namee.Text + "', Отчество = '" + Otch.Text + "', Дата\_рождения = '" + Date.Text + "', Код\_класса = " + Convert.ToInt32(Klass.Text) + " WHERE Код = " + Convert.ToInt32(Kod.Text) + ";";  
 MySqlCommand cmd = new MySqlCommand(upd, conn);  
 cmd.ExecuteNonQuery();  
 conn.Close();  
 }  
 catch (Exception exception)  
 {  
 Console.Write("Error" + exception);  
 }  
 }  
}

На остальных таблицах код добавления и редактирования сделан аналогичным способом.

### Фильтрация и поиск данных

Администратору доступны такие элементы управления как поиск, который выбирает и отображает подходящие под введенный текст услуги. Выборка нужных записей происходит в реальном времени, а также учитывает регистр введенного текста.

На рисунке 23 изображен функционал поля для поиска.

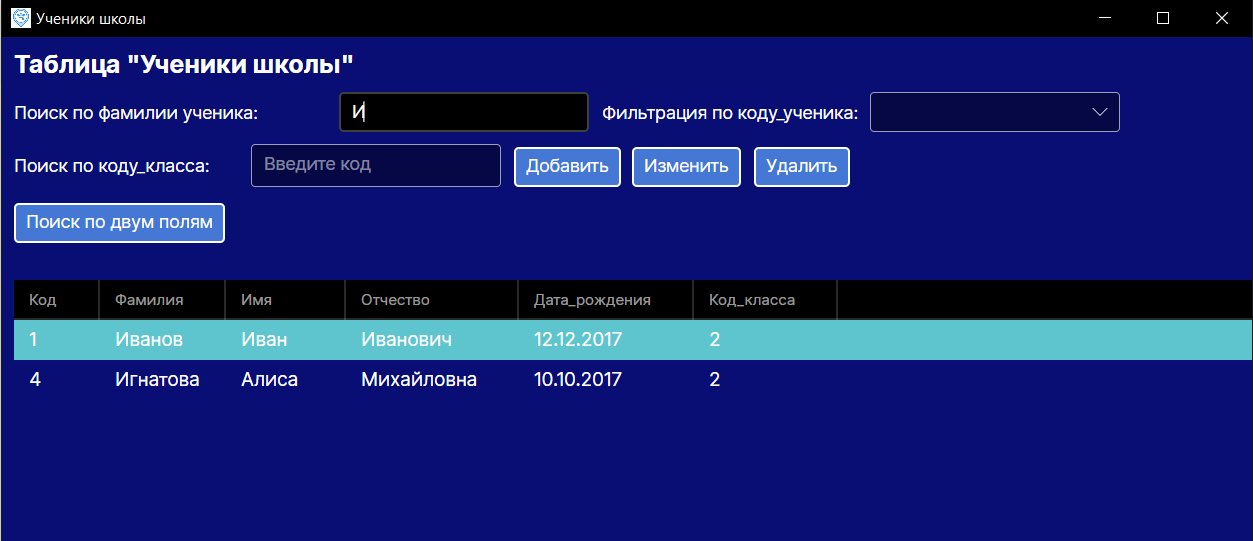


Рисунок 23 – Процесс поиска

Также на окне присутствует элемент выпадающего списка, который позволяет выполнить фильтрацию. Эта фильтрация позволит выбрать определенный список учеников, по тому или иному уникальному коду ученика.

На рисунке 24 изображен процесс фильтрации.

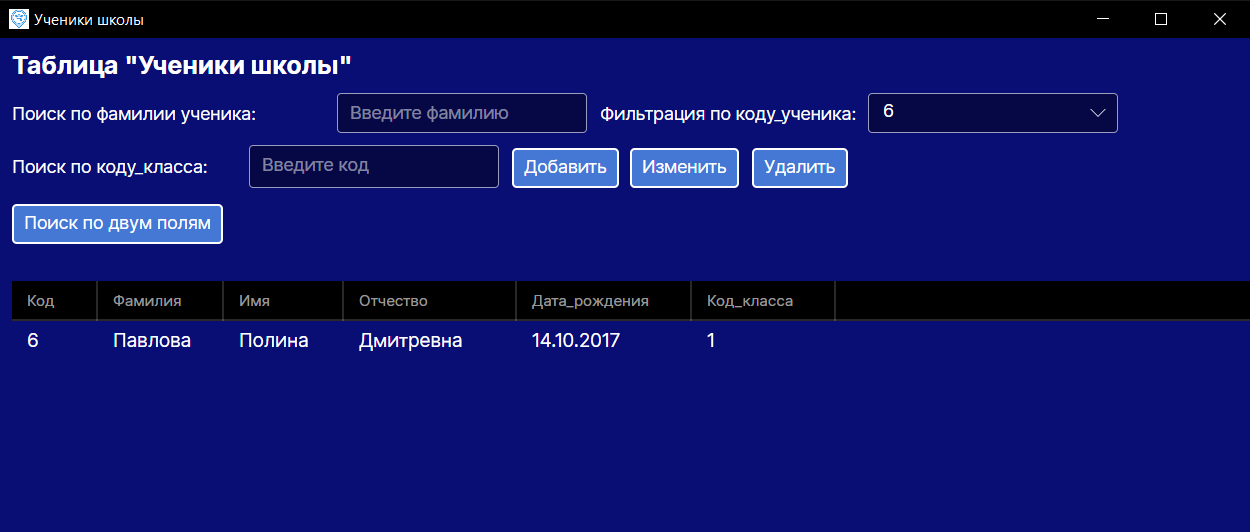


Рисунок 24 – Процесс фильтрации

# Тестирование

## Smoke-тестирование

Smoke-тестирование — это один из видов тестирования ПО, при котором выполняется набор базовых, но очень важных тестовых сценариев на ранних стадиях разработки продукта. Это позволяет быстро определить, готова ли система к более детальному и глубокому тестированию. Другими словами, smoke-тестирование — это проверка основных функций приложения на «готовность к работе».

Smoke-тестирование помогает:

* Быстро выявить критические дефекты, которые делают систему неработоспособной;
* Сэкономить время и ресурсы, не тратя их на более детальное тестирование неработоспособной системы;
* Ускорить процесс разработки и тестирования, так как сразу видны результаты работы.

Smoke-тестирование будет производиться в программе, а именно на таблице «кружки». В данном тестировании важен функционал добавления данных в таблицу.

Smoke-тестирование будет производиться по следующим пунктам:

* Откроем программу и войдем под учетной записью администратора;
* Перейдем на страницу, где необходимо добавить данные в таблицу;
* Введем данные для добавления в таблицу;
* Посмотрим, добавились ли данные в таблицу;
* Проверим, правильно ли отображаются данные в таблице;
* Попробуем удалить данные из таблицы и убедимся, что они удалились;
* Убедимся, что программа не выдает ошибок или работает некорректно.

На рисунке 24 изображена таблица в изначальном состоянии.

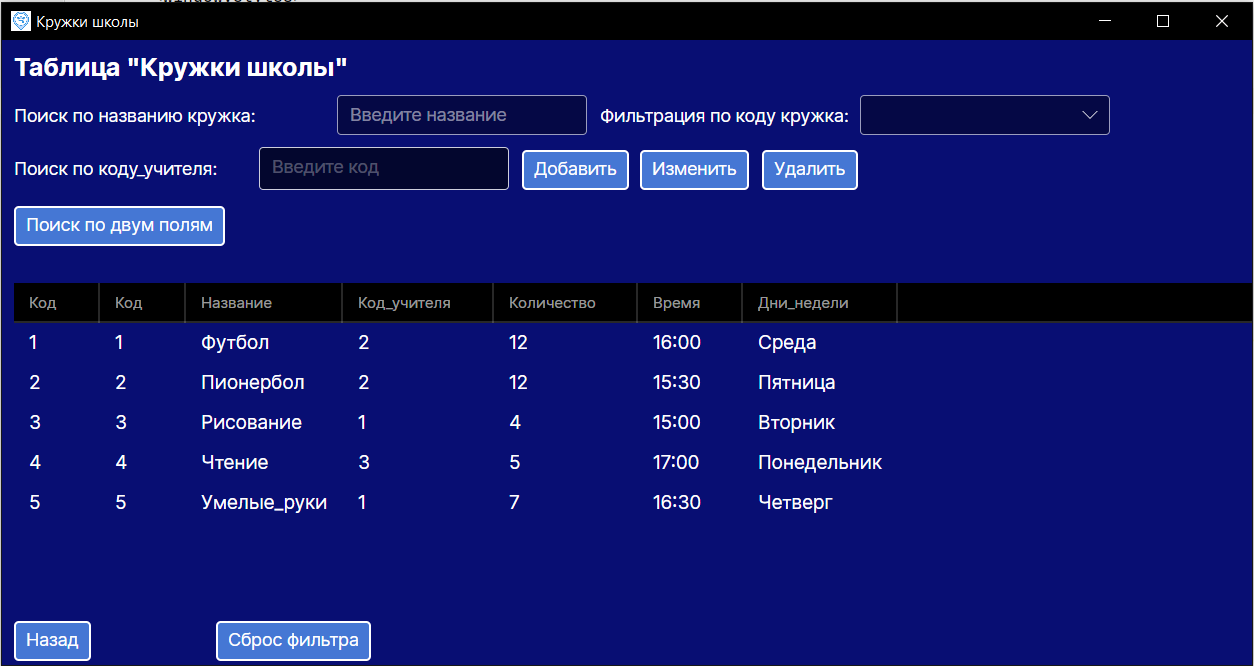


Рисунок 24 – Таблица в изначальном состоянии

На рисунке 25 изображено добавление данных.

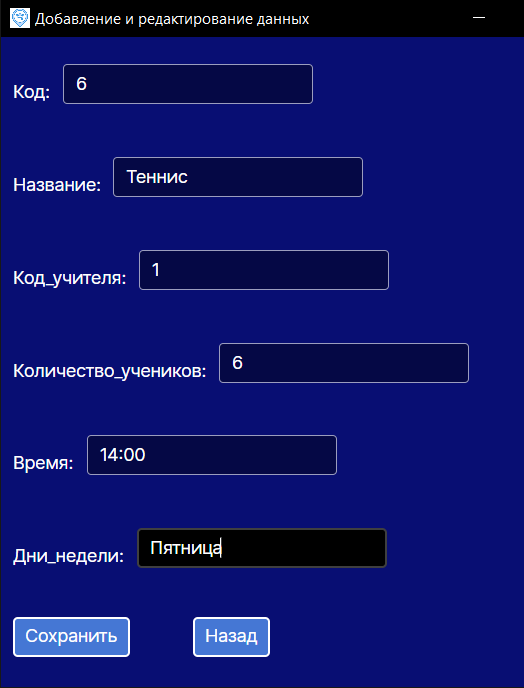


Рисунок 25 – Добавление данных

После добавления данных необходимо проверил правильность добавления данных, для того чтобы убедиться в корректности всех данных.

На рисунке 26 изображен результат добавления.

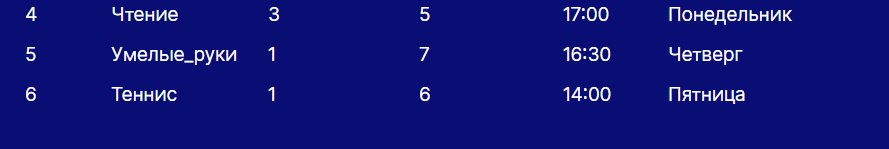


Рисунок 26 – Результат добавления

После того как убедились что данные добавились корректно произведем удаление этих же самых данных при помощи выделения записи и нажатия на кнопку «Удалить».

На рисунке 27 изображен результат удаления данных.

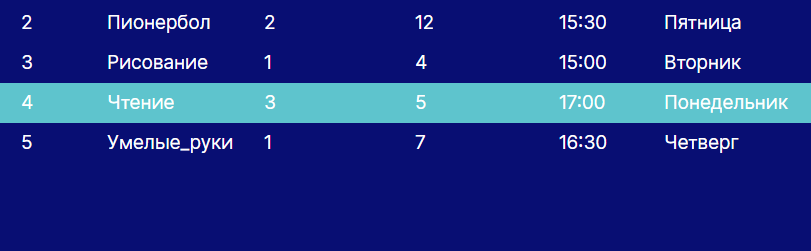


Рисунок 27 – Результат удаления

После удаления данные пропали из таблицы приложения и самой БД. Поэтому можем сделать вывод о том, что программа работает корректно и не выдает никаких ошибок при добавлении и последующем удалении данных.

## Тест-кейсы

Таблица 1- Тест-кейс №1 «Авторизация пользователя».

|  |  |
| --- | --- |
| Номер | Тест-кейс №1 |
| Название | Авторизация пользователя |
| Подготовка к тесту | Предусловие:  Войти в учетную запись с правами пользователя, логин:client и пароль: client |
| Шаги тестовой ситуации | Шаги:   1. Запустить программу школы. 2. Перейти на страницу авторизации. 3. Войти под учетной записью пользователя. |
| Результат выполнения тест-кейса | Ожидаемый результат:  Вход будет выполнен успешно.  Фактическийрезультат:  Вход выполнен успешно. |
| Возвращение системы в исходное состояние | Постусловия:  Если вход с правами пользователя выполнен успешно, то выйти из программы. |

Таблица 2- Тест-кейс №2 «Авторизация администратора».

|  |  |
| --- | --- |
| Номер | Тест-кейс №2 |
| Название | Авторизация администратора |
| Подготовка к тесту | Предусловие:  Войти в учетную запись с правами администратора, логин:admin и пароль: admin |
| Шаги тестовой ситуации | Шаги:   1. Запустить программу школы. 2. Перейти на страницу авторизации. 3. Войти под учетной записью администратора. |
| Результат выполнения тест-кейса | Ожидаемый результат:  Вход будет выполнен успешно.  Фактическийрезультат:  Вход выполнен успешно. |
| Возвращение системы в исходное состояние | Постусловия:  Если вход с правами администратора выполнен успешно, то выйти из программы. |

Таблица 3- Тест-кейс №3 «Отображение данных в качестве таблиц из БД».

|  |  |
| --- | --- |
| Номер | Тест-кейс №3 |
| Название | Отображение данных в качестве таблиц из БД |
| Подготовка к тесту | Предусловие:  Отобразить данные в качестве таблиц из БД |
| Шаги тестовой ситуации | Шаги:   1. Запустить программу школы. 2. Войти под учетной записью любого пользователя. 3. Перейти на страницу, на которой отображена таблица. 4. Проверить отображение данных и их корректность. |

Продолжение таблицы 3- Тест-кейс №3 «Отображение данных в качестве таблиц из БД».

|  |  |
| --- | --- |
| Результат выполнения тест-кейса | Ожидаемый результат:  Таблица отображается и данные корректны.  Фактический результат:  Таблица отобразилась и данные корректны. |
| Возвращение системы в исходное состояние | Постусловия:  Выполнить выход из программы. |

Продолжение таблицы 4- Тест-кейс №4 «Добавление данных в таблицу и БД».

|  |  |
| --- | --- |
| Номер | Тест-кейс №4 |
| Название | Добавление данных в таблицу и БД |
| Подготовка к тесту | Предусловие:  Проверить добавления данных через программу |
| Шаги тестовой ситуации | Шаги:   1. Запустить программу школы. 2. Войти под учетной записью администратора. 3. Перейти на страницу, на которой отображена таблица и необходимо, добавить данные. 4. Нажать на кнопку «Добавить» и перейти на специальную форму с добавлением 5. Заполнить все данные и нажать «Сохранить» 6. Вернуться назад и проверить добавление данных и их корректность |
| Результат выполнения тест-кейса | Ожидаемый результат:  Данные добавятся в таблицу, и они будут корректны.  Фактический результат:  Данные добавились в таблицу, и они корректны. |
| Возвращение системы в исходное состояние | Постусловия:  Если данные добавились, то удалить их при помощи кнопки «Удалить данные» |

Продолжение таблицы 5- Тест-кейс №5 «Поиск данных в таблице».

|  |  |
| --- | --- |
| Номер | Тест-кейс №5 |
| Название | Поиск данных в таблице |
| Подготовка к тесту | Предусловие:  Проверить поиск данных |
| Шаги тестовой ситуации | Шаги:   1. Запустить программу школы. 2. Войти под учетной записью администратора. 3. Перейти на страницу, на которой отображена таблица и необходимо, выполнить поиск. 4. Ввести в поле поиска текст 5. Проверить результат с набранным текстом |

Продолжение таблицы 5- Тест-кейс №5 «Поиск данных в таблице».

|  |  |
| --- | --- |
| Результат выполнения тест-кейса | Ожидаемый результат:  Поиск должен работать исправно и с правильным результатом.  Фактический результат:  Поиск работает исправно и выводит правильные результаты. |
| Возвращение системы в исходное состояние | Постусловия:  Очистить поле для поиска при помощи кнопки «Сброс фильтра» |

# Вывод

В ходе работы была рассмотрена предметная область, выбраны средства разработки, и хранения данных. В приложении получилось реализовать добавление, удаление, редактирования БД через платформу C#. Благодаря различным элементам управления стало возможно осуществлять фильтрацию, поиск данных в таблицах БД.

Таким образом, цель производственной практики, которая заключалась в создании и администрировании базы данных по учету информации об организации школы при помощи платформы C# достигнута.

Выполнены следующие задачи:

* Настройка и проектирование систем управления баз данных;
* Реализация клиентского приложения;
* Использование и реализация объектов баз данных;
* Освоение администрирования базы данных.

Выводы о проделанной работе могут быть следующими:

Успешное добавление данных в таблицу программы при smoke-тестировании свидетельствует о правильной работе функционала добавления данных. Это гарантирует, что приложение корректно обрабатывает информацию и сохраняет ее в базе данных.

Возможные сферы внедрения программного средства включают в себя:

* Автоматизацию учета успеваемости учеников и преподавателей;
* Улучшение коммуникации между участниками образовательного процесса;
* Обеспечение доступа к образовательным материалам и заданиям онлайн.

Перспективы развития программного средства могут включать в себя:

* Расширение функционала для включения дополнительных возможностей;
* Интеграцию с другими образовательными платформами и сервисами;
* Развитие мобильного приложения для удобного доступа к функционалу на мобильных устройствах.

Таким образом, приложение может значительно улучшить образовательный процесс и облегчить взаимодействие между участниками образовательного процесса, что делает его перспективным и востребованным программным средством в области образования.

# Приложение

# Приложение А. Программный код в системе контроля версий