

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

КУРСОВАЯ РАБОТА

**Разработка многопользовательской автоматизированной
системы управления организацией.**

Объект автоматизации – Школа

по дисциплине «Базы данных»

Выполнил
студент гр. 33534/5



Стойкоски Н.С.

Руководитель

Вишневская Т. А.

« ____ » _____ 2018 г.

Санкт-Петербург
2018

ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)
студенту группы 33534/1 Стойкоски Никола
(номер группы) (фамилия, имя, отчество)

1. Тема проекта (работы) Разработка многопользовательской
автоматизированной системы управления организацией. Задание № 1

Объект автоматизации – школа

2. Срок сдачи студентом законченного проекта (работы) 25.12.2018

3. Исходные данные к проекту (работе)

Описание предметной области

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов: введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения).

Введение. Анализ предметной области. Проектирование схемы данных.

Реализация базы данных в среде SQL Server. Разработка представлений и хранимых процедур. Разработка клиентского приложения. Тестирование.

Заключение. Список использованных источников.

Примерный объем пояснительной записки 15-20 страниц машинописного текста

5. Перечень графического материала (с указанием обязательных чертежей и плакатов) не предоставляется

6. Консультанты

7. Дата получения задания: « 06 » сентября 2018 г.

Руководитель

(подпись)

Винникова Т.А.
(инициалы, фамилия)

Задание принял к исполнению

(подпись)

Стойкоски Н.С.
(инициалы, фамилия)

13.09.2018
(дата)

Содержание

Введение.....	4
Анализ предметной области	5
Проектирование схемы данных.....	11
Реализация базы данных в среде SQL Server.....	12
Разработка представлений и хранимых процедур.....	14
Разработка клиентского приложения	18
Заключение.....	23
Список использованной литературы	24

Введение

Данный продукт представляет собой система работы с данных в школе, реализованную на основе Microsoft SQL Server 2014 с использованием C++/Qt. Эта база данных позволяет более эффективно управлять данными. Продукт может быть применен в школу для автоматизация управления и контроллирования данных.

Система предназначена для разные типы пользователи: Завуч, учитель, ученик.

Функциональность системы:

- Создание новых пользователей.
- Просмотр личных данных.
- Просмотр/изменение списки – учеников, учителей, групп, занятий
- Просмотр/изменение учебного плана, расписания, журнала
- Просмотр/изменение повышение квалификации учителей.
- Получение дополнительных информации – как на пример: список учеников у которых средняя оценка, меньше 4

Анализ предметной области

Группы пользователей разрабатываемой информационной системы (ИС)

№ пп	Наименование пользователя
1	Директор
2	Завуч
3	Преподаватель
4	Ученик

Функции групп пользователей

№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
<i>Первый пользователь(группа) Завуч</i>				
1	Добавление информации об учителе	Имя, фамилия, отчество, год рождения, дата поступления на работы, преподаваемый предмет,	Добавление информации об учителе в БД	Открытие окна с информацией об учителе и элементами для его редактирования
2	Удаление выбывшего учителя	Имя, фамилия, отчество	Удаление информации об учителе из БД	Открытие окна с информацией об учителе и элементами для его редактирования

3	Добавление и редактирование повышение квалификации и достижения учителя	Дата получения, тип, описание	Добавление информации об повышении квалификации и в БД	Открытие окна с информацией о квалификации и достижении учителя и элементами для его редактирования
4.	Составление и редактирование учебного плана	Класс, предметная область, учебный предмет, количество часов в неделю	Добавление информации об учебный план в БД	Открытие окна с информацией о учебного плана и элементами для его редактирования
5.	Добавление информации об ученике	Имя, фамилия, отчество, год рождения, класс	Добавление информации об ученике в БД	Открытие окна с информацией об ученике и элементами для его редактирования
6.	Удаление выбывшего ученика	Имя, фамилия, отчество, класс	Удаление информации об ученике из БД	Открытие окна с информацией об ученике и элементами для его редактирования
7.	Формирование расписания	День класс, название предмета, время, преподаватель, аудитория	Добавление информации об расписании в БД	Открытие окна с расписанием и элементами для его редактирования

8.	Редактирование расписания	Класс, название предмета, день недели	Добавление информации об расписании в БД	Открытие окна с расписанием и элементами для его редактирования
9.	Назначение классных руководителей	Преподаватель, номер класса	Добавление информации об классного руководителя в БД	Открытие окна с информацией об учителе и элементами для его редактирования
Третий пользователь(группа) Учитель				
1	Проставление оценки ученика в журнале	Имя, фамилия, отчество, номер ученика, класс, предмет, оценка	Добавление информации оценки ученика в БД	Открытие окна с формой проставления оценки, с предварительным выбором предмета
2	Просмотр расписания	Номер преподавателя	Вывод расписания на экран	Открытия окна с информацией о расписании на день
4	Просмотр журнала своего класса	Класс	Вывод журнала класса на экран	Открытия окна со журнала класса.
Четвертый(группа) Ученик				
1	Просмотр расписания	Класс	Вывод расписания на экран	Открытия окна с информацией о расписании

2	Просмотр оценок в журнале	Номер ученика	Вывод оценок на экран	Открытие окна с оценками
3.	Просмотр своих личные данные	Номер ученика	Вывод личных данных на экран	Открытие окна с личных данных

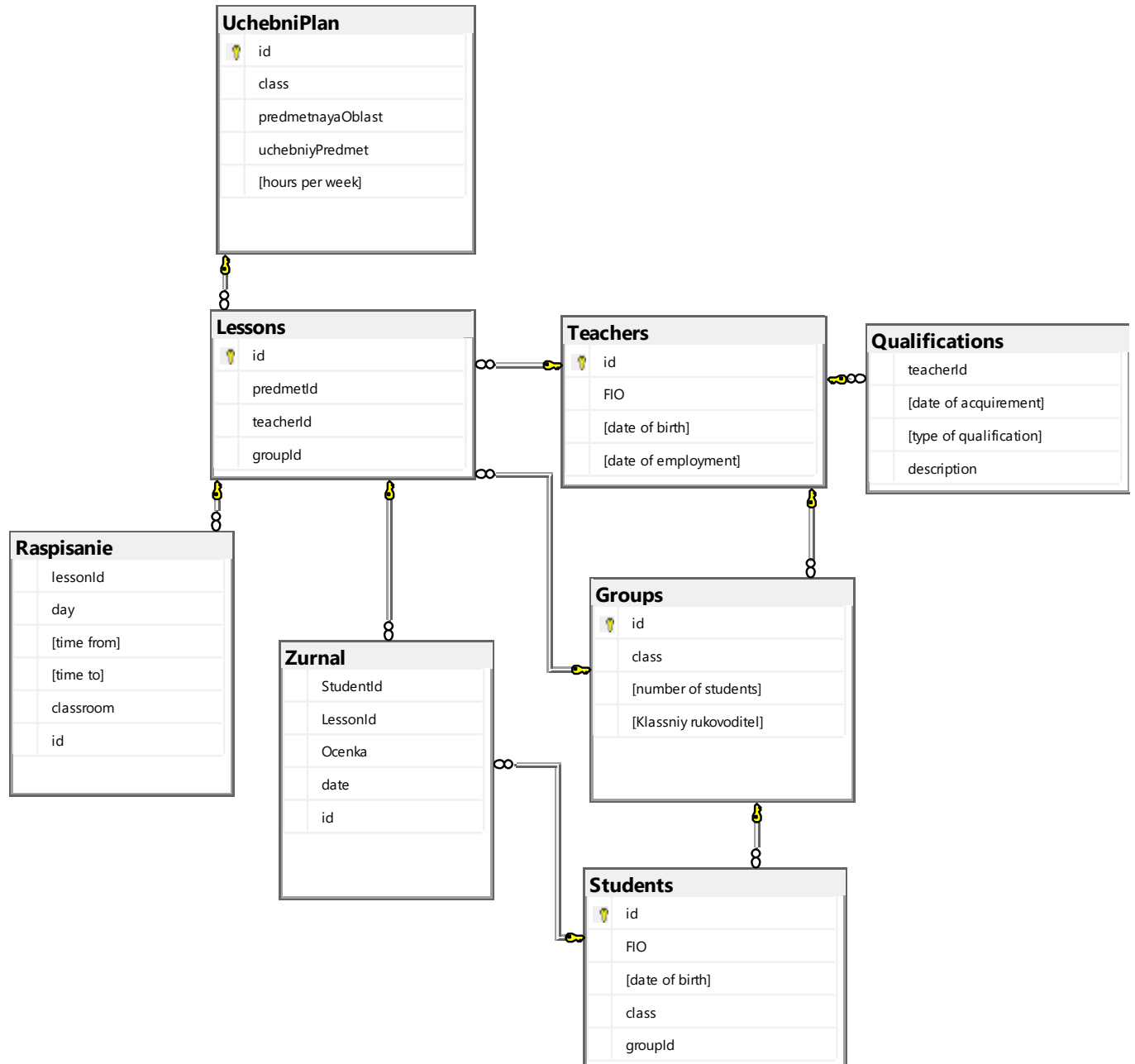
Хранимые данные

№ пп	Хранимые данные	Пользователи, которым разрешен доступ	Ограничения по типу и значению
1	Преподаватель ID ФИО Дата рождения Дата поступления на работу	Директор, завуч преподаватель	int char(30) date date
2	Ученик ID ФИО Дата рождения Класс Группа	Директор, завуч, преподаватель, ученик	int char(50) date smallint int
3	Группа ID класс	Директор, завуч, преподаватель, ученик	int smallint

	Количество учеников Классный руководитель		smallint int
4	Журнал ID ученика ID занятия Оценка Дата	Директор, завуч, преподаватель, ученик	int int smallint date
5	Расписание ID занятия День Время од Время до Аудитория	Директор, Завуч, Преподаватель, ученик	int tinyint time time char(50)
6	Повышение квалификации и достижения ID преподавателя Дата получения Тип Описание	Директор	int Date Char(30) Text
7	Учебный план ID Класс Предметная область	Директор, завуч, преподаватель, ученик	int smallint char(30)

	Учебный предмет Количество часов в неделю		char(30) smallint
8	Занятия ID ID предмета ID преподавателя ID группы	Директор, завуч, преподаватель, ученик	int int int int

Проектирование схемы данных



Реализация базы данных в среде SQL Server

```
use School2
create database School2
```

1. Создание таблицы учителя.

```
create table Teachers(
    id int IDENTITY(1,1) primary key not null,
    FIO char(50) not null,
    [date of birth] date not null,
    [date of employment] date not null
)
```

2. Создание таблицы повышение квалификации

```
create table Qualifications(
    teacherId int FOREIGN KEY REFERENCES Teachers(id) not null,
    [date of acquirement] date not null,
    [type of qualification] char(60),
    [description] text not null
)
```

3. Создание таблицы группы

```
create table Groups(
    id int IDENTITY(1,1) primary key not null,
    class smallint,
    [number of students] smallint not null DEFAULT 0,
    [Klassniy rukovoditel] int FOREIGN KEY REFERENCES Teachers(id)
)
```

4. Создание таблицы студентов

```
create table Students(
    id int IDENTITY(1,1) primary key not null,
    FIO char(50) not null,
    [date of birth] date not null,
    class smallint,
    groupId int FOREIGN KEY REFERENCES Groups(id)
)
```

5. Создание таблицы учебный план

```
create table UchebniPlan(
    id int IDENTITY(1,1) primary key not null,
    class smallint not null,
    predmetnayaOblast char(60) not null,
    uchebniyPredmet char(60) not null,
    [hours per week] smallint not null
)
```

6. Создание таблицы занятий

```
create table Lessons(  
    id int IDENTITY(1,1) primary key not null,  
    predmetId int REFERENCES UchebniPlan(id) not null,  
    teacherId int REFERENCES Teachers(id) not null,  
    groupId int REFERENCES Groups(id) not null  
)
```

7. Создание таблицы расписание

```
create table Raspisanie(  
    lessonId int REFERENCES Lessons(id) not null,  
    [day] tinyint not null CHECK ([day] > 0 and [day] < 7),  
    [time from] time(7) not null,  
    [time to] time(7) not null,  
    classroom char(30)  
)
```

8. Создание таблицы журнал

```
create table Zurnal(  
    StudentId int REFERENCES Students(id) not null,  
    LessonId int REFERENCES Lessons(id) not null,  
    Ocenka tinyint not null,  
    [date] date not null DEFAULT GETDATE()  
)
```

9. Триггеры для определения количество студентов в группу.

```
CREATE TRIGGER add_student_group ON Students  
FOR INSERT  
AS  
BEGIN  
    DECLARE @id_group INT  
    SELECT @id_group = INSERTED.groupId FROM INSERTED  
    UPDATE Groups SET [number of students] += 1 WHERE id = @id_group  
END
```

```
CREATE TRIGGER update_student_group ON Students  
FOR UPDATE  
AS  
BEGIN  
    DECLARE @id_group_old INT  
    DECLARE @id_group_new INT  
    SELECT @id_group_old = DELETED.groupId FROM DELETED  
    SELECT @id_group_new = INSERTED.groupId FROM INSERTED  
    UPDATE Groups SET [number of students] -= 1 WHERE id = @id_group_old  
    UPDATE Groups SET [number of students] += 1 WHERE id = @id_group_new  
END
```

```
CREATE TRIGGER delete_student_group ON Students  
FOR DELETE  
AS
```

```

BEGIN
    DECLARE @id_group INT
    SELECT @id_group = DELETED.groupId FROM DELETED
    UPDATE Groups SET [number of students] -= 1 WHERE id = @id_group
END

```

Разработка представлений и хранимых процедур

1. Занятия которые есть у студента

```

Create Procedure Student_Lessons @StudentId int as
Begin
    Select UchebniPlan.uchebniyPredmet, Teachers.FIO
    from Lessons, Students, Teachers, UchebniPlan
    where Students.id = @StudentId and Lessons.groupId = Students.groupId and
        Lessons.teacherId = Teachers.id and Lessons.predmetId = UchebniPlan.id
end

```

2. Расписание для студента

```

Create Procedure Student_Raspisanie @StudentId int as
begin
    select day, [time from], [time to], classroom, UchebniPlan.uchebniyPredmet,
    Teachers.FIO, Lessons.groupId
    from Raspisanie, Students, Teachers, Lessons, UchebniPlan where
        Students.id = @StudentId and Students.groupId = Lessons.groupId and Teachers.id =
        Lessons.teacherId and
        UchebniPlan.id = Lessons.predmetId and Raspisanie.lessonId = Lessons.id
    order by day, [time from]
end

```

3. Журнал для студента

```

Create Procedure Student_Zurnal @StudentFIO char(50) as
begin
    select uchebniyPredmet, Ocenka, Teachers.FIO, Zurnal.date
    from Zurnal, Students, Lessons, UchebniPlan, Teachers where
        Students.FIO = @StudentFIO and Zurnal.StudentId = Students.id and
        Zurnal.LessonId = Lessons.id and UchebniPlan.id = Lessons.predmetId and
        Teachers.id = Lessons.teacherId
end

```

4. Занятия для группа

```

Create Procedure Group_Lessons @GroupId int as
begin
    select UchebniPlan.uchebniyPredmet, Teachers.FIO
    from Lessons, UchebniPlan, Teachers where Lessons.groupId = @GroupId and
        Lessons.predmetId = UchebniPlan.id and Lessons.teacherId = Teachers.id
end

```

5. Расписание для группа

```

Create Procedure Group_Raspisanie @GroupId int as
begin
    select day, [time from], [time to], classroom, UchebniPlan.uchebniyPredmet,
    Teachers.FIO, Lessons.groupId
    from Raspisanie, Teachers, Lessons, UchebniPlan where
        Lessons.groupId = @GroupId and Teachers.id = Lessons.teacherId and

```

```

        UchebniPlan.id = Lessons.predmetId and Raspisanie.lessonId = Lessons.id
        order by day, [time from]
end

```

6. Журнал для группа

```

Create Procedure Group_Zurnal @GroupId int as
begin
    select Students.FIO, uchebniyPredmet, Ocenka, Teachers.FIO, Zurnal.date
    from Zurnal, Students, Lessons, UchebniPlan, Teachers where
        Students.groupId = @GroupId and Zurnal.StudentId = Students.id and
        Zurnal.LessonId = Lessons.id and UchebniPlan.id = Lessons.predmetId and
        Teachers.id = Lessons.teacherId
end

```

7. Занятия у учителя

```

Create Procedure Teacher_Lessons @FIO char(50) as
begin
    select UchebniPlan.uchebniyPredmet, Lessons.groupId, Groups.class
    from Lessons, UchebniPlan, Groups, Teachers where Teachers.FIO = @FIO and
        Teachers.id = Lessons.teacherId and Lessons.predmetId = UchebniPlan.id
        and Groups.id = Lessons.groupId
end

```

8. Расписание для учителя

```

Create Procedure Teacher_Raspisanie @FIO char(50) as
begin
    select day, [time from], [time to], classroom, UchebniPlan.uchebniyPredmet,
    Lessons.groupId
    from Raspisanie, Teachers, Lessons, UchebniPlan where Teachers.FIO = @FIO
        and Teachers.id = Lessons.teacherId and UchebniPlan.id = Lessons.predmetId
        and Raspisanie.lessonId = Lessons.id
        order by day, [time from]
end

```

9. Средняя оценка ученика

```

Create Procedure Student_Avg_Mark @FIO char(50) as
begin
    Select avg(CAST(Zurnal.Ocenka as float)) as [Avg. Ocenka]
    from Zurnal, Students where Students.FIO = @FIO
        and Zurnal.StudentId = Students.id
end

```

10. Средняя оценка для всех учеников в группу

```

Create Procedure Group_Student_Avg @GroupId int as
begin
    select Students.FIO, avg(cast(Zurnal.Ocenka as float)) as [Avg. Ocenka]
    from Zurnal, Students where Students.groupId = @GroupId and Zurnal.StudentId =
    Students.id
    group by Students.FIO
    order by [Avg. Ocenka]
end

```

11. Средняя оценка всех групп по каждому предмету

```

Create Procedure Group_Predmet_Avg @GroupId int as
begin

```

```

    select UchebniPlan.uchebniyPredmet, cast(avg(cast(Zurnal.Ocenka as float)) as
decimal(10,2)) as [Avg. Ocenka]
  from UchebniPlan, Lessons, Zurnal where Lessons.groupId = @GroupId
    and Zurnal.LessonId = Lessons.id and Lessons.predmetId = UchebniPlan.id
 group by UchebniPlan.uchebniyPredmet order by [Avg. Ocenka]
end

```

12. Список групп

```

Create View [List of Groups] as
Select Groups.id, class, [number of students], Teachers.FIO as [Классный руководитель]
  from Groups, Teachers
    where Teachers.id = Groups.[Klassniy rukovoditel]

```

13. Список занятий

```

Create View [List of Lessons] as
Select UchebniPlan.uchebniyPredmet, Teachers.FIO, Lessons.groupId
  from Lessons, UchebniPlan, Teachers
    where UchebniPlan.id = Lessons.predmetId and Teachers.id = Lessons.teacherId

```

14. Количество студентов

```

Create View [Total number of students] as
select sum(Groups.[number of students]) as [Total Students] from Groups

```

15. Средняя оценка всех студентов

```

Create View [Avg Student Marks] as
select Students.FIO, avg(cast(Zurnal.Ocenka as float)) as [Avg. Ocenka]
  from Zurnal, Students where Students.id = Zurnal.StudentId group by Students.FIO
 order by [Avg. Ocenka]

```

16. Средняя оценка всех групп

```

Create View [Avg Group Marks] as
select a.groupId, Groups.[number of students], a.[Avg. Ocenka], Teachers.FIO as [Классный
руководитель]
  from
    (select Students.groupId, cast(avg(cast(Zurnal.Ocenka as float)) as decimal(10,2))
as [Avg. Ocenka]
    from Zurnal, Students where Zurnal.StudentId = Students.id group by
Students.groupId) a,
    Groups, Teachers
    where a.groupId = Groups.id and Groups.[Klassniy rukovoditel] = Teachers.id

```

17. Средняя оценка от каждого учителя

```

Create View [Avg Teacher Marks] as
select Teachers.FIO, cast(avg(cast(Zurnal.Ocenka as float)) as decimal(10,2)) as [Avg.
Ocenka]
  from Zurnal, Lessons, Teachers where Zurnal.LessonId = Lessons.id and Lessons.teacherId
= Teachers.id
 group by Teachers.FIO order by [Avg. Ocenka]

```

18. Средняя оценка каждой группы по каждому предмету

```

Create View [Avg mark for Group - Predmet] as
select Groups.class, a.uchebniyPredmet, a.groupId, a.[Avg. Ocenka] from
    (select Lessons.groupId, UchebniPlan.uchebniyPredmet,
    cast(avg(cast(Zurnal.Ocenka as float)) as decimal(10,2)) as [Avg. Ocenka]
    from UchebniPlan, Lessons, Zurnal where Zurnal.LessonId = Lessons.id

```



```

        and Lessons.predmetId = UchebniPlan.id group by Lessons.groupId,
        UchebniPlan.uchebniPredmet) a,
        Groups where Groups.id = a.groupId order by Groups.class, a.uchebniPredmet, a.[Avg.
        Ocenka]

```

19. Количество занятий в день для каждой группы

```

Create View [Group - Number of lessons per day] as
select Raspisanie.Day, Lessons.groupId, count(Raspisanie.lessonId) as [Number of Lessons]
from Raspisanie, Lessons
where Raspisanie.lessonId = Lessons.id
group by Raspisanie.Day, Lessons.groupId
order by Raspisanie.day

```

20. Количество занятий в день для каждого учителя

```

Create View [Teacher - Number of lessons per day] as
select Raspisanie.Day, Teachers.FIO, count(Raspisanie.lessonId) as [Number of Lessons]
from Raspisanie, Lessons, Teachers where Raspisanie.lessonId = Lessons.id and
Lessons.teacherId = Teachers.id group by Raspisanie.day, Teachers.FIO
order by Raspisanie.day

```

21. Количество занятий в неделю для каждого учителя

```

Create View [Teacher - Number of lessons per week] as
select Teachers.FIO, count(Raspisanie.lessonId) as [Number of Lessons]
from Raspisanie, Teachers, Lessons where Raspisanie.lessonId = Lessons.id and
Lessons.teacherId = Teachers.id group by Teachers.FIO order by [Number of Lessons]

```

22. Количество учеников в каждой группы, имеющие среднюю оценку меньше 4.

```

Create View [Group - Number of Students with avg < 4.0] as
select Students.groupId, count(a.StudentId) as [Students with AVG < 4.0] from
(select StudentId, cast(avg(cast(Zurnal.Ocenka as float)) as decimal(10,2)) as [avg]
from Zurnal group by StudentId) a,
Students where Students.id = a.StudentId and a.avg < 4.0 group by Students.groupId

```

23. Список учеников имеющие среднюю оценку < 4

```

Create View [List of students with avg < 4.0] as
select Students.class, Students.groupId, Students.FIO, a.avg from
(select Zurnal.StudentId, cast(avg(cast(Zurnal.Ocenka as float)) as decimal(10,2)) as [avg]
from Zurnal group by Zurnal.StudentId) a,
Students where Students.id = a.StudentId and avg < 4.0
order by Students.class, a.avg

```

24. Количество занятий в расписанию и количество занятий по учебному плану, для каждой группы

```

Create View [Raspisanie <-> Uchebniy Plan] as
select b.GroupId, b.[Lessons in uchebni plan per week], ISNULL(c.numLessons, 0) as
[Lessons in raspisanie per week]
from
(select Groups.id as GroupId, a.[Lessons in uchebni plan per week] from
(select UchebniPlan.class, sum(UchebniPlan.[hours per week]) as [Lessons in uchebni
plan per week]
from UchebniPlan group by UchebniPlan.class) a,
Groups where Groups.class = a.class) b
FULL OUTER JOIN

```

```

(select Groups.id, count(Raspisanie.lessonId) as [numLessons] from Raspisanie,
Lessons, Groups
where Groups.id = Lessons.groupId and Lessons.id = Raspisanie.lessonId group by
Groups.id) c
ON b.GroupId = c.id

```

25. Количество оценок и количество предметов для каждого ученика

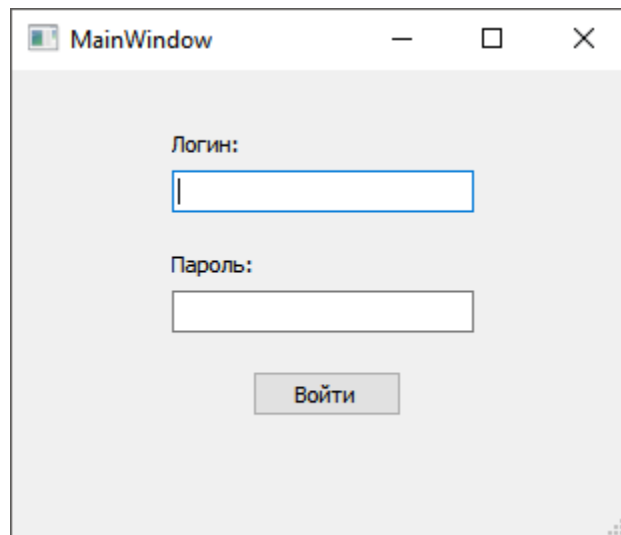
```

Create View [Student - Kol.Predmetov - Kol.Ocenok] as
select a.FIO, a.[Kol. predmetov], b.[Kol. ocenok] from
(select Students.FIO, count(predmetId) as [Kol. predmetov] from Lessons, Students where
Students.groupId = Lessons.groupId group by Students.FIO) a,
(select Students.FIO, count(Zurnal.StudentId) as [Kol. ocenok] from Zurnal, Students
where Students.id = Zurnal.StudentId group by Students.FIO) b
where a.FIO = b.FIO order by b.[Kol. ocenok]

```

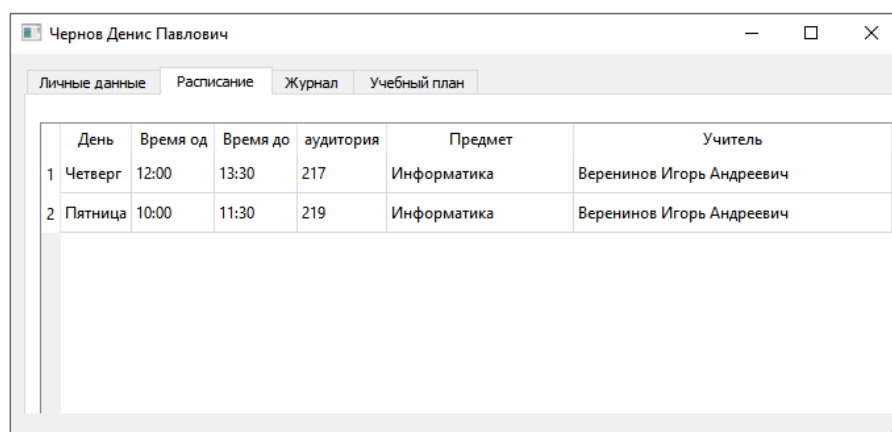
Разработка клиентского приложения

1. Начальное окно



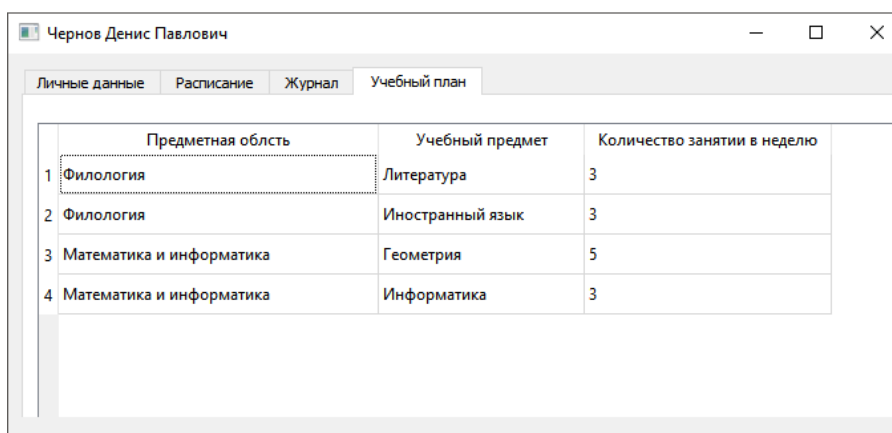
Пользователь вводит свой логин и пароль, если успешно то соответствующее ему окно открывается, в другом случае выводится сообщение “Invalid username or password!”.

2. Режим работы ученика



The screenshot shows a window titled 'Чернов Денис Павлович' with four tabs: 'Личные данные', 'Расписание', 'Журнал', and 'Учебный план'. The 'Расписание' tab is active, displaying a table with the following data:

	День	Время од	Время до	аудитория	Предмет	Учитель
1	Четверг	12:00	13:30	217	Информатика	Веренинов Игорь Андреевич
2	Пятница	10:00	11:30	219	Информатика	Веренинов Игорь Андреевич



The screenshot shows the same window with the 'Учебный план' tab active. It displays a table with the following data:

	Предметная область	Учебный предмет	Количество занятий в неделю
1	Филология	Литература	3
2	Филология	Иностранный язык	3
3	Математика и информатика	Геометрия	5
4	Математика и информатика	Информатика	3

Ученик может посмотреть свой личные данные, посмотрееть свое расписание, свои оценки, поставленные в журнале, и свой учебный план.

3. Режим работы учителя

Веренинов Игорь Андреевич

Личные данные Расписание Журнал

Предмет Информатика Группа 5

	Ученик	Оценка	Дата
1	Петров Денис Петрович	5	12/11/2018
2	Нгуен Хай Йен	5	12/11/2018
3	Соколова Александра Димитровна	4	12/11/2018

Вставить запись Удалить запись Редактировать запись

Веренинов Игорь Андреевич

Личные данные Расписание Журнал

Предмет Информатика Группа 5

	Ученик	Оценка	Дата
1	Петров Денис Петрович	5	12/11/2018
2	Нгуен Хай Йен	5	12/11/2018
3	Соколова		

Dialog

Ученик
Петров Денис Петрович
Круглов Андрей Андреевич
Соколова Александра Димитровна

Оценка

Дата
2018-12-20

Сохранить Отмена

Вставить запись Удалить запись Редактировать запись

Учитель может посмотреть свои личные данные и свое расписание. Также может проставлять оценок в журналу для своих групп и предметов.

4. Режим работы завуча

MainWindow

Студенты Группы Учители Учебный план Занятия Расписание Дополнительно

Сортировка -

	Студент	Дата рождения	Класс	Группа
1	Алейников Павел Алексеевич	3/24/1998	9	1
2	Бойков Кирилл Валентинович	10/17/1998	11	4
3	Артемова Виктория Димитровна	1/1/1997	11	4
4	Шкригунов Алексей Константинович	11/13/1997	11	4
5	Максимов Валентин Кирилович	5/23/1997	11	4
6	Петров Денис Петрович	4/23/1998	11	5
7	Круглов Андрей Андреевич	6/29/1997	11	5
8	Соколова Александра Димитровна	4/5/1998	11	5
9	Нгуен Хай Йен	11/5/1997	12	12
10	Чернов Денис Павлович	10/9/1997	12	12

Вставить запись Удалить запись Редактировать запись

MainWindow

Студенты Группы Учители Учебный план Занятия Расписание Дополнительно

Сортировка -

	День	Время од	Время до	аудитория	Предмет	Учитель	группа
1	Понедельник	10:00	11:30	217	Физика	Попов Борис Петрович	4
2	Понедельник	10:00	11:30	217	Информатика	Веренинов Игорь Андреевич	1
3	Вторник	10:00	11:30	219			
4	Среда	10:00	11:30	192a			
5	Среда	12:00	13:30	217			
6	Четверг	10:00	11:30	219b			
7	Четверг	12:00	13:30	217			
8	Четверг	12:00	13:30	237a			
9	Четверг	14:00	15:30	237a			
10	Пятница	10:00	11:30	219			

Вставить запись Удалить запись Редактировать запись

Dialog

День Понедельник

Время од 10:00

Время до 10:45

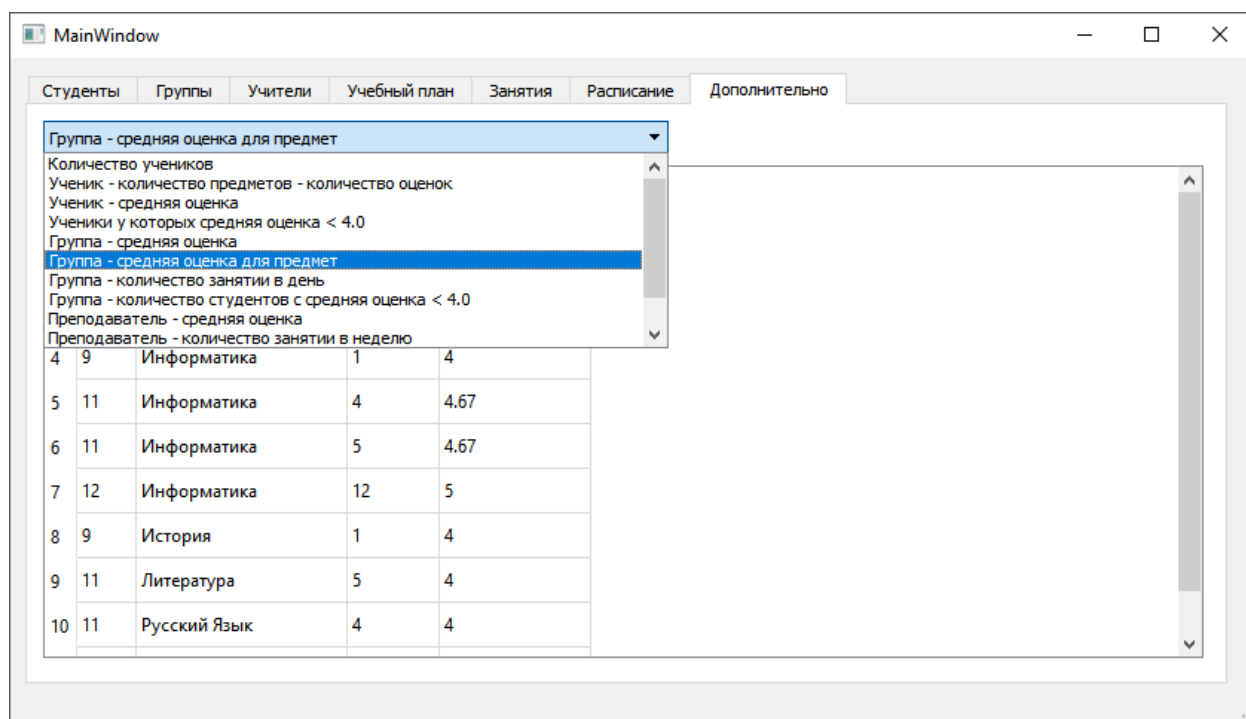
аудитория 217

Предмет Физика

Учитель Попов Борис Петрович

группа 5

Сохранить Отмена



Завуч может добавлять/изменять/удалять данные об учеников, группы, учителя. Может менять учебный план. Может создавать создавать занятий, и их проставлять в расписанию. Так же может смотреть дополнительные таблицы.

Заключение

В ходе курсовой работы была реализована система база данных для школы в Microsoft SQL Server 2017, а так же и клиентское приложение написано на языке C++ с использованием графической библиотеки Qt.

Созданная база данных и клиентское приложение, позволяют более эффективно управлять данными. Продукт может быть применен в школу для автоматизация управления и контролирования данных.

Список использованной литературы

1. Березин Б.И., Березин С.Б. Начальный курс С и С++. М.: ДИАЛОГ-МИФИ, 1996
2. А. Перерва, В. Иванова, Путь аналитика. Практическое руководство IT-специалиста, - Москва, Издательство: Питер, 2015 г., 282 стр.
3. Боэм, Б и др. Характеристики качества программного обеспечения / Б. Боэм, Дж. Брун, Х. Гаспар и др. - М.: Мир, 1981 - 420 с.
4. Боэм, Б. Инженерное проектирование программного обеспечения / Б. Боэм. - М.: Радио и связь, 1985 - 234 с.
5. Ван Тассел Д. Стил, разработка, эффективность, отладка и испытание программ. — М.:Мир, 1981