Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

Направление подготовки программная инженерия

Профиль беcпрофильный

**КУРСОВОЙ ПРОЕКТ**

по дисциплине Базы данных

(наименование дисциплины)

на тему: Информационная система «Школа XXI века»

(тема курсового проекта)

Выполнил студент   4  курса*\_\_\_\_\_\_*группы 16-КБ-ПР1

  Овезов Мырат Нургелдиевич

(ф.и.о.)

Допущен к защите\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель (нормоконтролер) работы   ст.преп. Ковтун А.А.

Защищен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии:   к.т.н., доцент Янаева М.В.

ст. преп. Носова Ю.С.

(подпись, дата, расшифровка подписи)

Краснодар

2019

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО КубГТУ)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

Направление подготовки программная инженерия

Профиль беспрофильный

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_\_\_\_\_М.В. Янаева

« »    2019 г.

**ЗАДАНИЕ**

на курсовой проект

Студенту Овезов Мырату Нургелдиевичу группы 16-КБ-ПР1 4 курса

(Ф.И.О.) (№ группы и курса)

Тема работы: «Информационная система «Школа XXI века»

(утверждена указанием директора института №38/19-кт от 17.09.2019 г).

План работы:

1. Изучение предметной области

2. Проектирование с использованием средств СУБД и разработки ПО

3. Описание реализованной модели

Объем работы:

а) пояснительная записка     с.

б) листинг     листа

Рекомендуемая литература

1. Белоногов Г.Г., Новоселов А.П. «Автоматизация процессов накопления, поиска и обобщения информации»

2. Ульман Дж. «Основы систем баз данных»

Срок выполнения: с «\_\_\_» \_\_\_\_\_\_по «\_\_\_»\_\_\_\_20\_\_г.

Срок защиты:                      «\_\_\_»\_\_\_\_20\_\_г.

Дата выдачи задания:                «\_\_\_»\_\_\_\_20\_\_г.

Дата сдачи работы на кафедру:       «\_\_\_»\_\_\_\_20\_\_г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   ст. преп. Ковтун А.А.

(должность, подпись,)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Овезов М.Н.

**Реферат**

Пояснительная записка курсового проекта: \_\_ страниц, \_\_ рисунков, 5 источников, 2 приложения.

БАЗЫ ДАННЫХ, СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ, СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ, УПРАВЛЕНИЕ ШКОЛОЙ, MY SQL SERVER, MARIADB, PHPMYADMIN.

Объектами разработки являются системы управления базами данных, а также автоматизированные системы учебного расписания.

Целью курсового проекта является закрепление основ и углубление знаний в области проектирования автоматизированных систем с внедренной в оную систему базы данных, а также разработка модели базы данных с последующей реализацией приложения, совместимого с данной базой.

К полученным результатам относятся модель базы данных в среде системы «Школа XXI века», а также приложение, полностью совместимое с имеющейся базой.

К решённым задачам относятся описание предметной области, концептуальное проектирование структур базы данных, реализация модели базы данных при помощи средств СУБД, а также реализация приложения автоматизированной системы.

Содержание

[**Введение** 5](#_Toc31897845)

[**1 Нормативные ссылки** 8](#_Toc31897846)

[**2 Описание предметной области** 9](#_Toc31897847)

[**2.1 Требования к структуре ПО и БД** 9](#_Toc31897848)

[**2.1.1 Функциональные требования** 9](#_Toc31897849)

[**2.1.2 Требования к интерфейсу** 9](#_Toc31897850)

[**2.1.3 Требования к базе данных** 10](#_Toc31897851)

[**3 Проектирование структуры базы данных** 11](#_Toc31897852)

[**3.1 Выявление сущностей** 11](#_Toc31897853)

[**3.2 Построение концептуальной модели** 13](#_Toc31897854)

[**4 Реализация базы данных средствами выбранной СУБД** 15](#_Toc31897855)

[**4.1 Выбор СУБД** 18](#_Toc31897856)

[**4.2 Реализация таблиц** 20](#_Toc31897857)

[**4.3 Построение ER- диаграммы** 23](#_Toc31897858)

[**5 Реализация приложения базы данных** 25](#_Toc31897859)

[**5.1 Выбор технологий реализации** 25](#_Toc31897860)

[**5.2 Реализация приложения** 26](#_Toc31897861)

[**Заключение** 34](#_Toc31897862)

[**Список использованных источников** 35](#_Toc31897863)

[Приложение А 36](#_Toc31897864)

[Приложение Б 37](#_Toc31897865)

## **Введение**

Реляционный способ доступа к данным основывается на операциях с группами записей. Для задания операций используются средства языка MySQL.

MySQL - это популярный сервер баз данных, используемый в разных приложениях. SQL означает язык структурированных запросов - (S)tructured (Q)uery (L)anguage, который MySQL использует для коммуникации с другими программами. Сверх того, MySQL имеет свои собственные расширенные функции SQL для того чтобы обеспечить пользователям дополнительный функционал.

Мир баз данных становится все более и более единым, что привело к необходимости создания стандартного языка, который мог бы использоваться, чтобы функционировать в большом количестве различных видов компьютерных сред. Стандартный язык позволит пользователям, знающим один набор команд, использовать их, чтобы создавать, отыскивать, изменять, и передавать информацию независимо от того работают ли они на персональном компьютере, сетевой рабочей станции, или на универсальной ЭВМ. Во все более и более взаимосвязанном компьютерном мире, пользователь, снабженный таким языком, имеет огромное преимущество в использовании и обобщении информации из ряда источников с помощью большого количества способов.

Элегантность и независимость от специфики компьютерных технологий, а также его поддержка лидерами промышленности в области технологии реляционных баз данных, сделало SQL, и вероятно в течение обозримого будущего оставит его, основным стандартным языком. По этой причине, любой, кто хочет работать с базами данных 90-х годов должен знать SQL.

Стандарт SQL определяется ANSI (Американским Национальным Институтом Стандартов) и в данное время также принимается ISO (международная организации по стандартизации). Однако, большинство коммерческих программ баз данных расширяют SQL без уведомления ANSI, добавляя разные другие особенности в этот язык, которые, как они считают, будут весьма полезны. Иногда они несколько нарушают стандарт языка, хотя хорошие идеи имеют тенденцию развиваться и вскоре становиться стандартами "рынка" сами по себе в силу полезности своих качеств. В этой книге, мы будем, в основном, следовать стандарту ANSI, но одновременно иногда будет показывать и некоторые наиболее общие отклонения от его стандарта.

Можно с большой степенью достоверности утверждать, что большинство приложений, которые предназначены для выполнения хотя бы какой-нибудь полезной работы, тем или иным образом используют структурированную информацию или, другими словами, упорядоченные данные. Такими данными могут быть, например, списки заказов на тот или иной товар, списки предъявленных и оплаченных счетов или список телефонных номеров ваших знакомых. Обычное расписание движения автобусов в вашем городе - это тоже пример упорядоченных данных.

При компьютерной обработке информации, упорядоченные каким- либо образом данные принято хранить в базах данных - особых файлах, использование которых вместе со специальными программными средствами позволяет пользователю, как просматривать необходимую информацию, так и, по мере необходимости, манипулировать ею.

*Цель курсового проекта* – закрепление основ и углубление знаний в области проектирования архитектуры и реализации базы данных, получение дополнительных практических навыков и изучение приемов разработки приложений с помощью Windows Forms.

## **1 Нормативные ссылки**

В данном документе использованы ссылки на следующие стандарты:

ГОСТ Р 1.5-2002 ГСС РФ. Стандарты. Общие требования к построению, изложению, оформлению, содержанию и обозначению

ГОСТ 7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления

ГОСТ 7.12-93 СИБИД. Библиографическая запись. Сокращения слов на русском языке. Общие требования и правила

ГОСТ 7.82-2001 СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления.

## **2 Описание предметной области**

База данных создаётся для обеспечения автоматизированного учёта процессов в Школе. БД должна реализовывать следующие функции:

* учёт учителей;
* учёт учеников;
* учёт предметов;
* учёт расписании;
* учёт классов;
* вывод различной информации в разных документах, необходимой для школы;

### **2.1 Требования к структуре ПО и БД**

### **2.1.1 Функциональные требования**

Программа должна формировать отчеты по выбранным данным и предоставлять их для печати.

### **2.1.2 Требования к интерфейсу**

Интерфейс должен быть предельно понятен для рядового пользователя, следовательно, он должен быть интуитивно понятным, а также чётко разграничивающим функциональность и варианты использования приложения.

### **2.1.3 Требования к базе данных**

Сущности, указанные в базе, должны чётко разграничивать свою функциональность, не иметь слишком много атрибутов, загромождающих память и не имеющих особого логического смысла.

Также, для первичных ключей рекомендуется ограничение по длине – 8 символов.

Тип данных для атрибутов, описывающих стоимость, ввиду практического отсутствия в денежном обороте копеек, рекомендуется установить, как целочисленный.

Ввиду потенциального принятия на работу иностранных граждан, рекомендуется установить атрибут, указывающий на отчество сотрудника, как необязательный к заполнению.

В качестве минимальных аппаратных (системных) требований утверждаются следующие:

* Процессор – Intel Core i3-3220 и выше;
* Частота процессора: 2 ГГц и выше;
* Объём ОЗУ: 3 ГБ и выше;
* Операционная система: Windows 7 и выше.

Программные требования:

Программа должна как можно детальнее описать процесс работы внутри автоматизированной системы школы (оформление прихода, ведение учёта учителей, учеников и др.).

Также, программа должна иметь авторизацию для входа в систему.

## **3 Проектирование структуры базы данных**

### **3.1 Выявление сущностей**

Сущность «Учитель» включающая в себя следующие атрибуты:

1. Код учителя;
2. Фамилия;
3. Имя;
4. Отчества;
5. Пол;
6. Дата рождение;
7. Адрес;
8. Электронная почта;
9. Телефон;
10. Логин;
11. Пароль;
12. Изображение учителя;

Сущность «Дисциплина», включающая в себя следующие атрибуты:

1. Название дисциплин;
2. Описание дисциплин.

Сущность «Ученик» включающая в себя следующие атрибуты:

1. Код ученика;
2. Имя ученика;
3. Фамилия ученика;
4. Отчество ученика;
5. Пол;
6. Дата рождения;
7. Адрес;
8. Телефон;
9. Название класса (Внешний ключ);
10. Логин;
11. Пароль;
12. Изображение ученика.

Сущность «Класс» включающая в себя следующие атрибуты:

1. Название класса;
2. Адрес класса (номер кабинета);
3. Количества учеников.

Сущность «Расписание» включающая в себя следующие атрибуты:

1. Код расписании;
2. День;
3. Начала времени;
4. Время окончании.

Сущность «Учитель с дисциплинами» включающая в себя следующие атрибуты:

1. Код учитель с дисциплинами;
2. Код учителя;
3. Код дисциплин.

Сущность «события», включающая в себя следующие атрибуты:

1. Код мастера;
2. Код учителя с дисциплиной;
3. Класс;
4. Расписание

### **3.2 Построение концептуальной модели**

Согласно пункту 3.1, имеется семь сущностей, которые необходимо связать в единую модель.

Сущность «Учитель» относится к сущности «Учитель с дисциплиной» как один ко многим, т.к у одного дисциплины один учитель, а у учителя много дисциплин. Также Сущность «Дисциплина» относится к сущности «Учитель с дисциплиной» но здесь все наоборот, т.к. тут у учителя может быть одна дисциплин может быть много учителей.

Сущность «Ученик» относится к сущности «Класс» как один ко многим, ведь один ученик имеет один класс, а у классов может быть много учеников.

Сущность «Расписание» относится к сущности «События» как многие ко многим, ведь одном уроке может быть много событии и в одном событии может быть много расписании.

Сущность события имеет отношении с сущностями «Учитель с дисциплинами», «Класс», «Расписание» многие ко многим, т.к. в одном событии может быть много классов, в одном классе может много учителя многими дисциплинами и соответственно в одном событии много расписании и т.д.

Конечная вариация концептуальной модели представлена на рисунке 1.

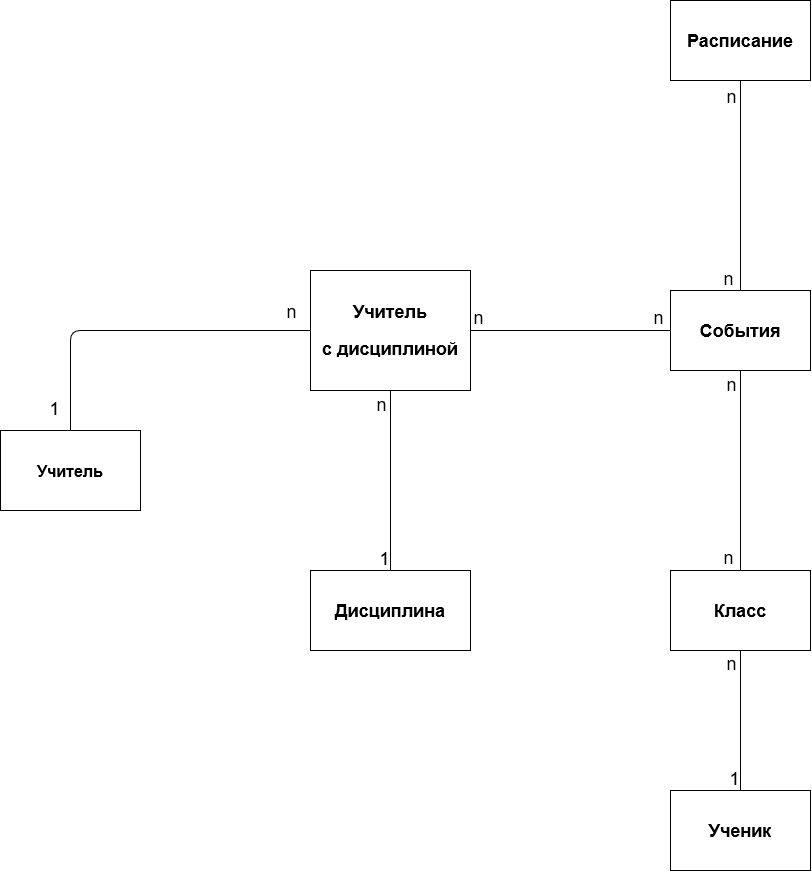


Рисунок 1 - Концептуальная модель

## **4 Реализация базы данных средствами выбранной СУБД**

СУБД управляет тремя важными вещами: данными, компонентом database engine (движок базы данных), который позволяет получать доступ к данным, блокировать и изменять их, и схемой базы данных, которая определяет логическую структуру базы данных. Эти три основополагающих элемента помогают обеспечить параллелизм, безопасность, целостность данных и единообразные процедуры администрирования данных. Типичные задачи администрирования баз данных, поддерживаемые СУБД, включают управление изменениями, мониторинг/настройку производительности, резервное копирование и восстановление. Многие системы управления базами данных также отвечают за автоматизированные откаты, перезапуск и восстановление, а также ведение журнала и аудит активности в базах данных.

СУБД, возможно, наиболее полезна для обеспечения централизованного представления данных, которые могут быть доступны нескольким пользователям из нескольких местоположений контролируемым образом. СУБД может ограничивать то, какие данные видит конечный пользователь, а также то, как этот конечный пользователь может просматривать данные, предоставляя множество представлений одной схемы базы данных. Конечным пользователям и программным программам не нужно понимать, где физически находятся данные или на каком типе носителей они находятся, поскольку СУБД обрабатывает все запросы.

СУБД может обеспечивать как логическую, так и физическую независимость данных. Это означает, что он может защитить пользователей и приложения от необходимости знать, где хранятся данные или беспокоиться об изменениях в физической структуре данных. До тех пор, пока программы используют интерфейс прикладного программирования (API) для базы данных, предоставляемой СУБД, разработчикам не придется изменять программы только потому, что в базу данных были внесены изменения.

В реляционной системе управления базами данных (СУБД), наиболее широко используемом типе СУБД, этот API является MYSQL, стандартным языком программирования для определения, защиты и доступа к данным в СУБД.

Использование СУБД для хранения и управления данными имеет свои преимущества, но и накладные расходы. Одним из самых больших преимуществ использования СУБД является то, что она позволяет конечным пользователям и прикладным программистам получать доступ и использовать одни и те же данные при управлении целостностью данных. Данные лучше защищены и поддерживаются, когда они могут совместно использоваться с помощью СУБД вместо создания новых итераций одних и тех же данных, хранящихся в новых файлах для каждого нового приложения. СУБД предоставляет центральное хранилище данных, доступ к которому может осуществляться несколькими пользователями контролируемым образом.

Централизованное хранение и управление данными в рамках СУБД обеспечивает:

* абстракция и независимость данных;
* безопасность данных;
* механизм блокировки для одновременного доступа;
* эффективный обработчик для балансировки потребностей нескольких приложений, использующих одни и те же данные;
* возможность быстрого восстановления после сбоев и ошибок, включая возможность перезапуска и восстановления;
* надежные возможности обеспечения целостности данных;
* ведение журнала и аудит деятельности;
* простой доступ с использованием стандартного API;
* единые процедуры администрирования данных.

### **4.1 Выбор СУБД**

My SQL Server - это система управления реляционными базами данных, разработанная корпорацией Oracle. Как сервер баз данных, это программный продукт с основной функцией хранения и извлечения данных по запросу других программных приложений—которые могут работать либо на том же компьютере, либо на другом компьютере по сети (включая интернет).

Корпорация Oracle выпускает по меньшей мере дюжину различных выпусков My SQL Server, предназначенных для различных аудиторий и рабочих нагрузок, начиная от небольших приложений на одной машине и заканчивая большими приложениями с выходом в Интернет и множеством одновременных пользователей. [2]

MySQL является решением для малых и средних приложений. Входить в состав серверов различных типов в том числе, WAMP (Windows Apache MySQL PHP), AppSery, LAMP (Linux Apache MySQL PHP), MAMP (Mac Apache MySQL PHP) и в разных портативных сборках таких как XAMPP, Денвер, VertigoServ, OpenSSH и т.д. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Благодаря своей мультиплатформенности, свободным реляционным данных MySQL является решением для многих разработчиков. Не смотря на свой такую большую популярность это СУБД тоже имеет свои недостатки и главным из них слабая защита, но опытные программисты легко исправляют ошибку в самом коде используя разнотипных символов.

Для удобств управление СУБД MySQL существует разные приложении такие как MySQL workbench в качестве десктопного варианта, PHPMYADMIN в качестве веб серверного варианта. Я выбрал PHPMYADMIN так как для меня более удобным.

### **4.2 Реализация таблиц**

Проектирование таблиц произведено со всеми нормами нормализации. Нормализация таблиц (отношений) — это формальный аппарат ограничений на формирование таблиц (отношений), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых в базе данных, уменьшает трудозатраты на ведение (ввод, корректировку) базы данных. Процесс нормализации заключается в разложении (декомпозиции) исходных отношений БД на более простые отношения. Каждая ступень этого процесса приводит схему отношений в последовательные нормальные формы. Для каждой ступени нормализации имеются наборы ограничений, которым должны удовлетворять отношения БД. Нормализация позволяет удалить из таблиц базы избыточную не ключевую информацию.

Структура БД изображена на рисунке 2.

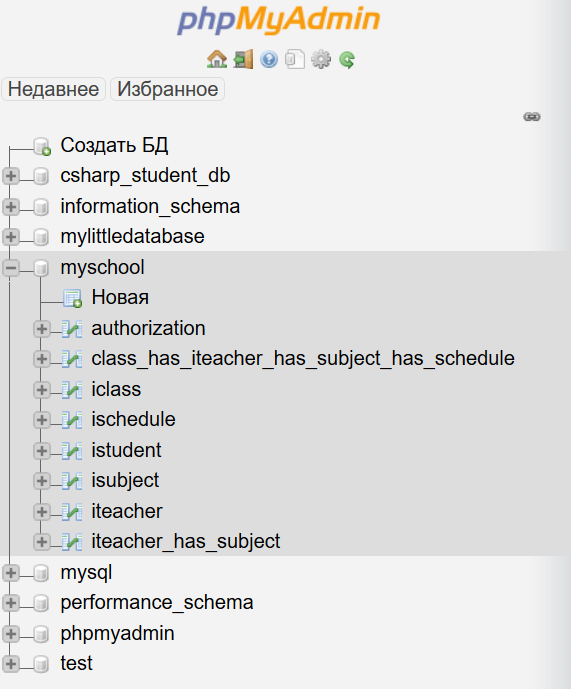


Рисунок 2 - Структура базы данных

Атрибуты таблиц базы данных «myschool» отображены на рисунке 3:

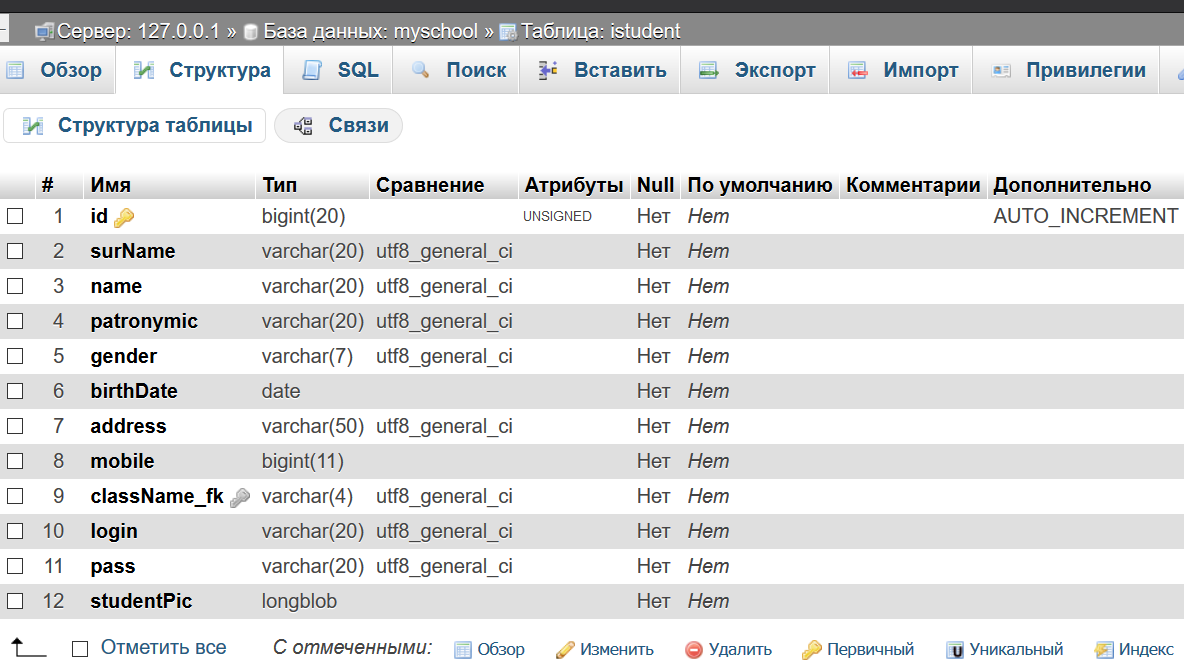


Рисунок 3 - Описание атрибутов таблицы istudent

### **4.3 Построение ER- диаграммы**

Модель сущностных отношений (или модель ER) описывает взаимосвязанные вещи, представляющие интерес в определенной области знаний. Базовая модель ER состоит из типов сущностей (которые классифицируют интересующие объекты) и определяет отношения, которые могут существовать между сущностями (экземплярами этих типов сущностей).

Сущность может быть определена как вещь, способная к независимому существованию, которое может быть однозначно идентифицировано. Сущность - это абстракция от сложностей предметной области. Когда мы говорим о сущности, мы обычно говорим о каком-то аспекте реального мира, который можно отличить от других аспектов реального мира.

Связь фиксирует, как сущности связаны друг с другом. Отношения можно рассматривать как глаголы, связывающие два или более существительных. ER – диаграмма базы «myschool» изображена на рисунке 7.

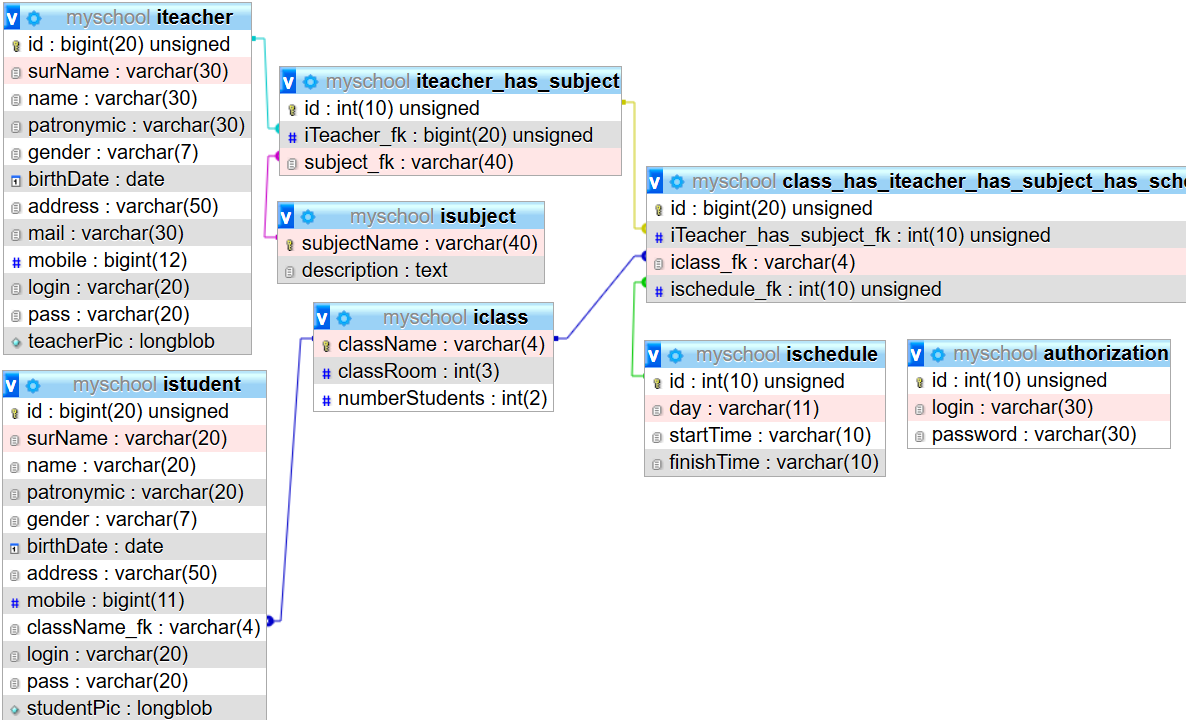


Рисунок 4 - Диаграмма базы данных

## **5 Реализация приложения базы данных**

### **5.1 Выбор технологий реализации**

Основным инструментом реализации является С# - это универсальный, многопарадигмальный язык программирования, охватывающий строгую типизацию, лексически ограниченные, императивные, декларативные, функциональные, универсальные, объектно-ориентированные (основанные на классах) и компонентно-ориентированные дисциплины программирования. Он был разработан около 2000 года корпорацией Майкрософт в рамках ее инициативы .NET, а затем утвержден в качестве международного стандарта Ecma (ECMA-334) и ISO (ISO/IEC 23270:2018). Mono-это название свободного проекта с открытым исходным кодом для разработки компилятора и среды выполнения для данного языка. C# является одним из языков программирования, предназначенных для инфраструктуры общего языка (CLI).

В реализации интерфейса нам поможет Windows Forms. Windows Forms (WinForms) - это графическая библиотека классов (GUI), входящая в состав Microsoft .NET Framework или Mono Framework, предоставляющая платформу для написания многофункциональных клиентских приложений для настольных, портативных и планшетных ПК. Хотя он рассматривается как замена более ранней и более сложной библиотеки классов Microsoft Foundation на основе C++, он не предлагает сопоставимой парадигмы и действует только как платформа для уровня пользовательского интерфейса в многоуровневом решении.

Для подключения MySQL к проекту и для выполнения SQL запросов нам понадобиться коннектор, коннектор можно скачать в официальном сайте MySQL и установить его в References изображена на рисунке 8.

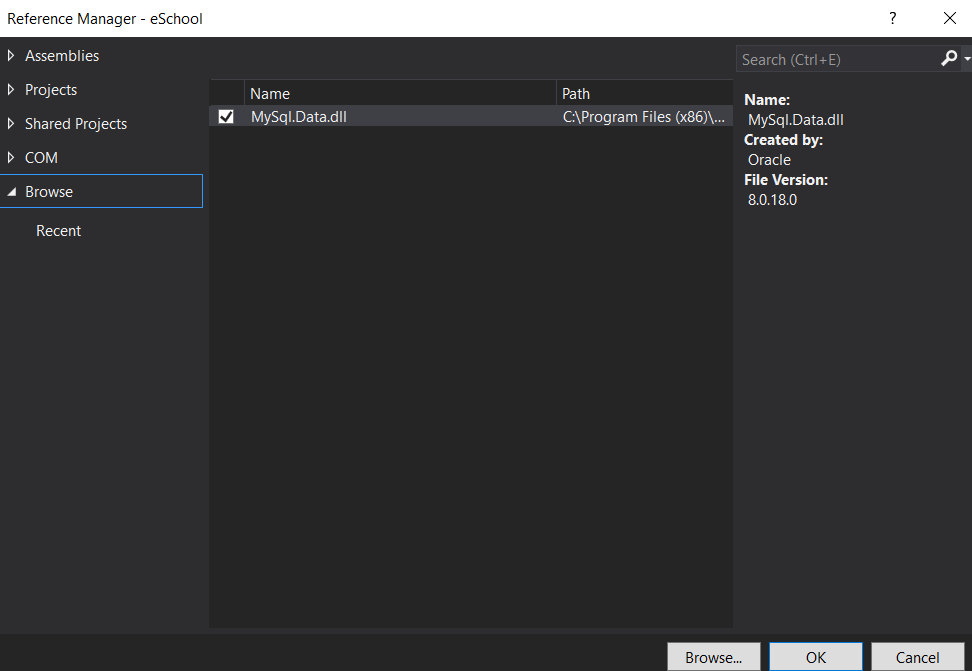


Рисунок 5 – Подключение MySQL в C#

Можно конечно и через ADO подключаться к базе, но это чревато тем что в будущем нам просто будет необходимо вносить изменения при переходе на Windows Vista и выше.

### **5.2 Реализация приложения**

Как только пользователь запускает программу на экране появиться окно авторизации (рис. 9). При успешной авторизации пользователь заходить в главное меню (рис 10.), где собственно говоря выполняется все функции системы.

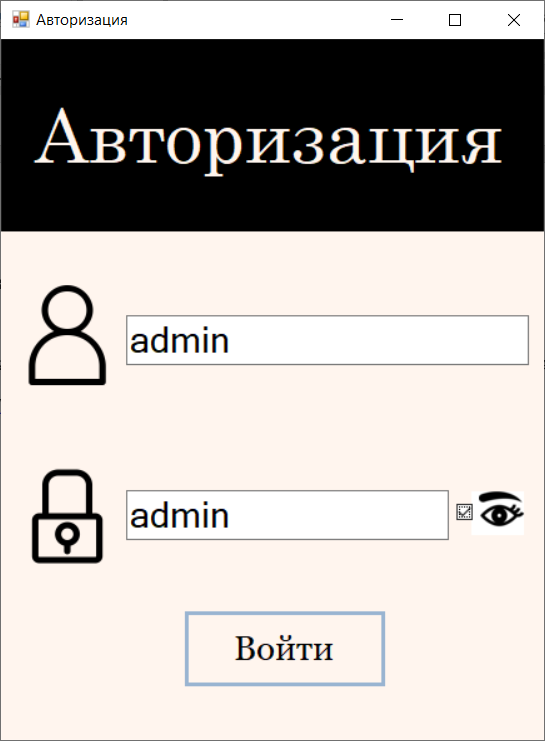


Рисунок 6 - Форма авторизации

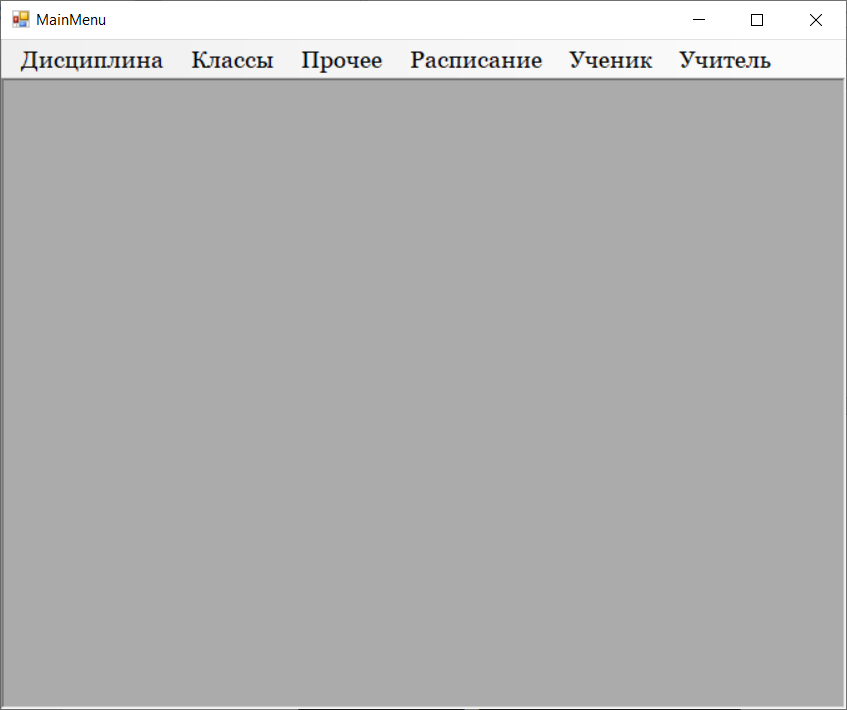


Рисунок 7 – Главное меню

Во всех кнопках в окне меню есть все необходимые функции такие как:

* Добавить (рис. 11)
* Изменить (рис. 12)
* Удалить (рис. 12)
* Управлять (рис. 13)
* Сортировать (рис. 14)
* Поиск (рис. 15) и т.д.

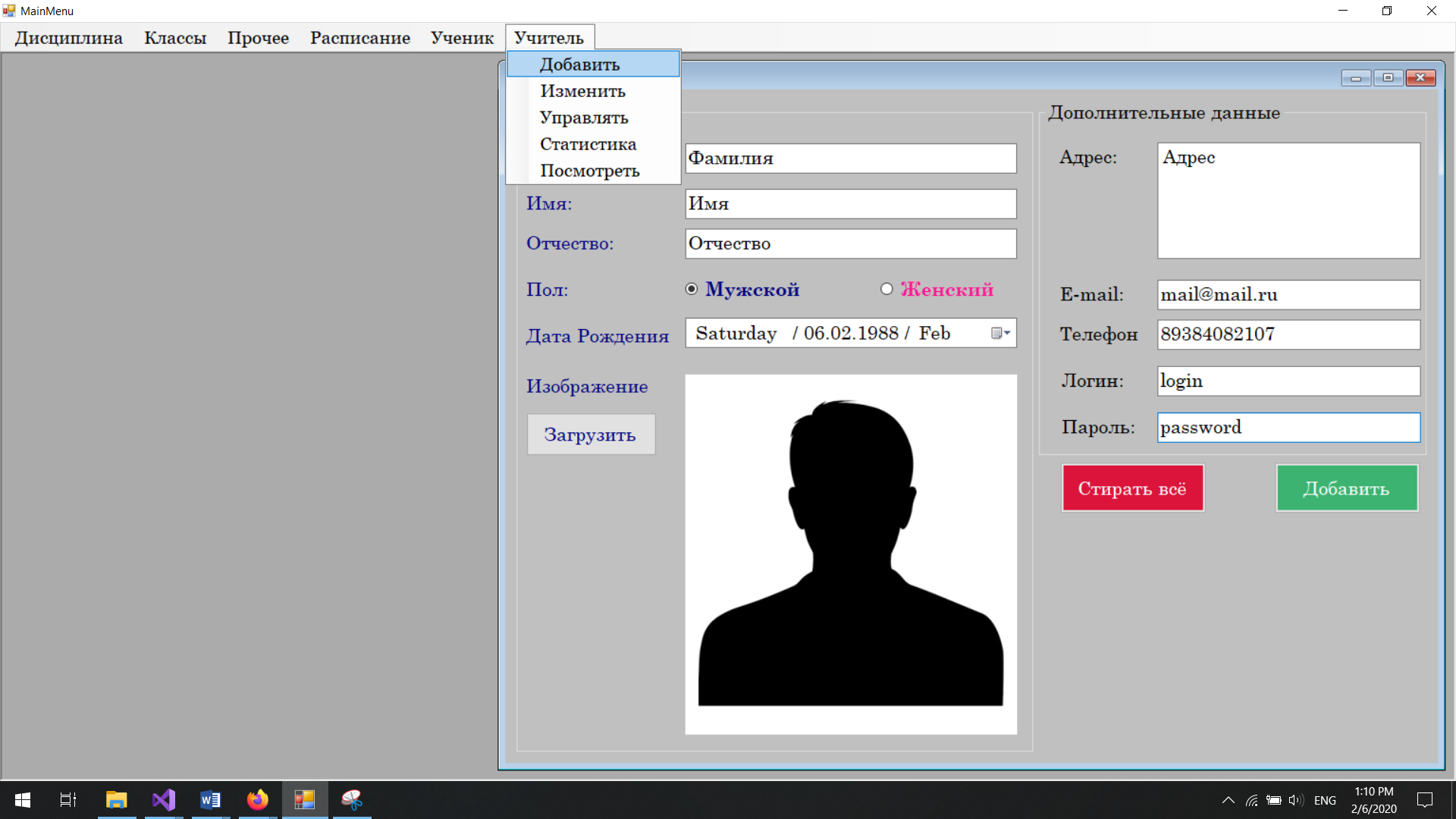
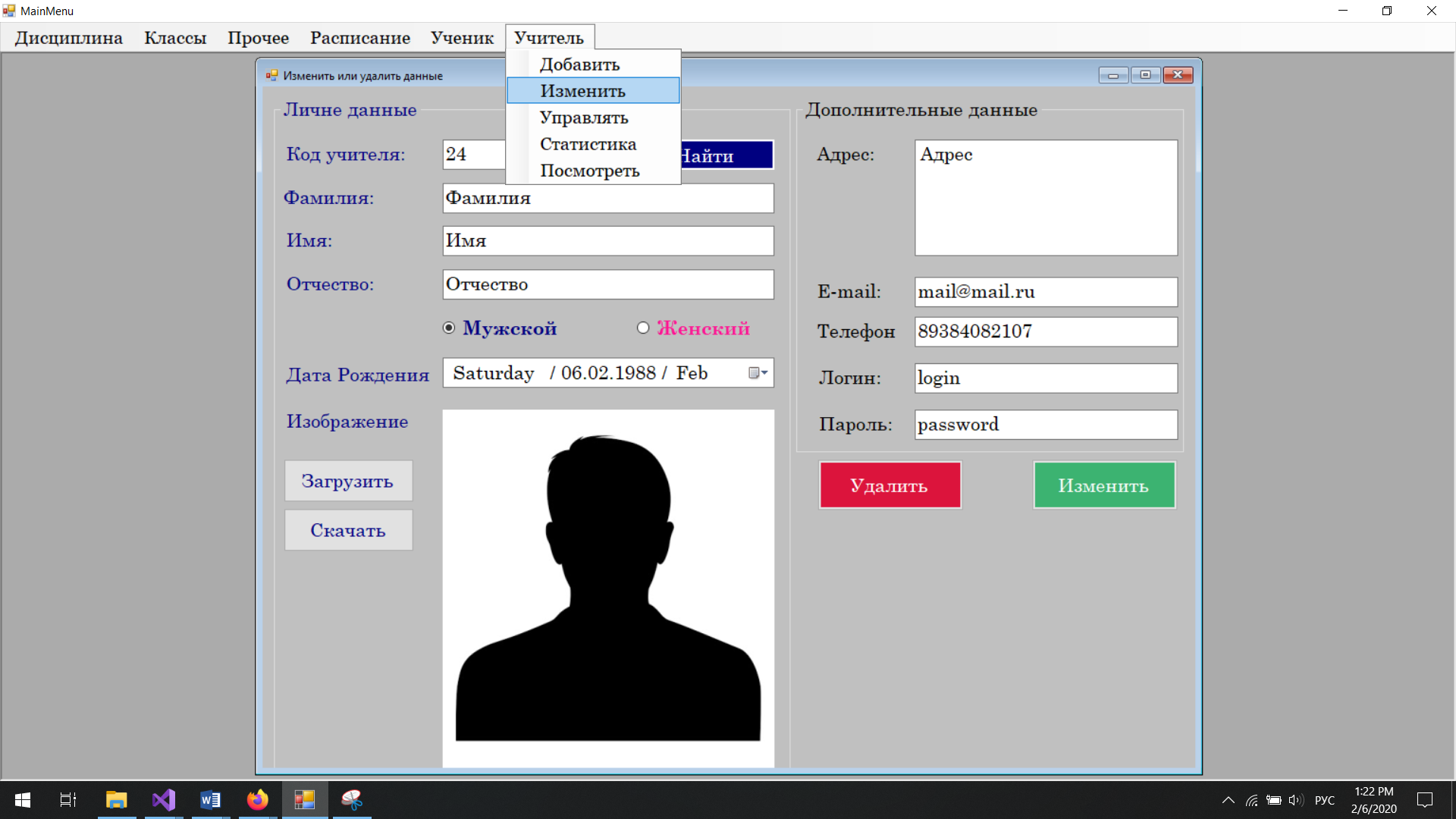
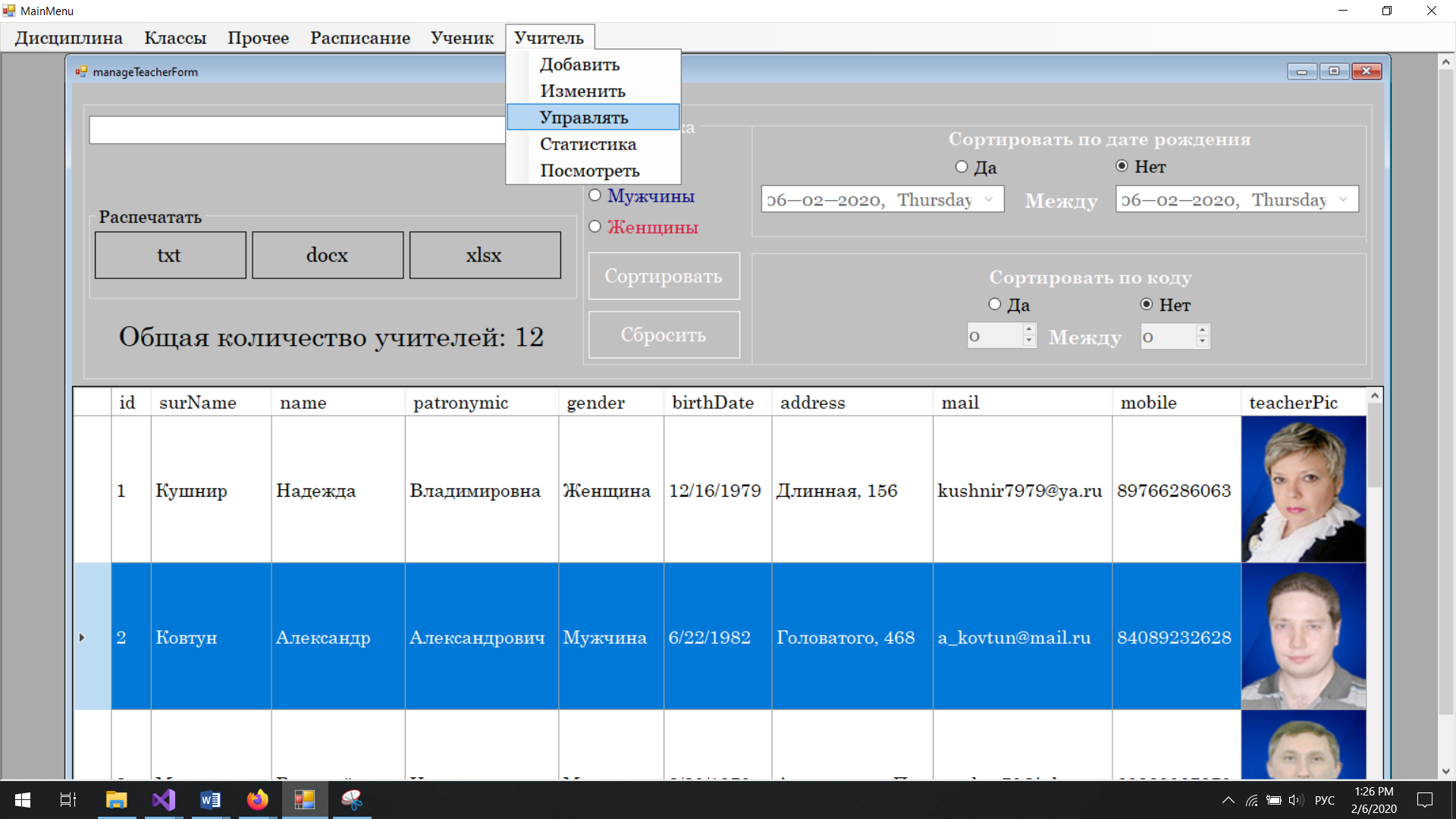
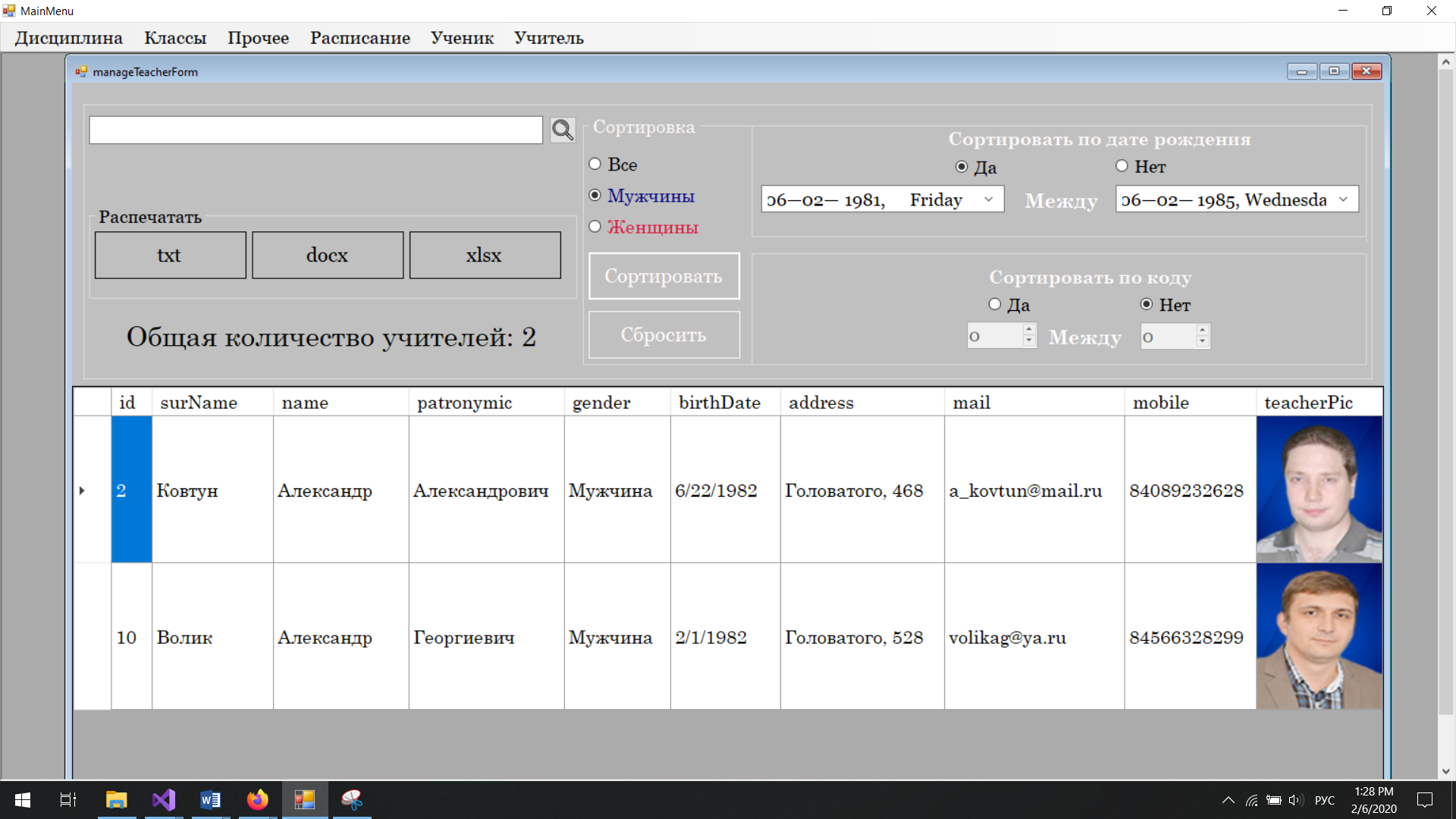
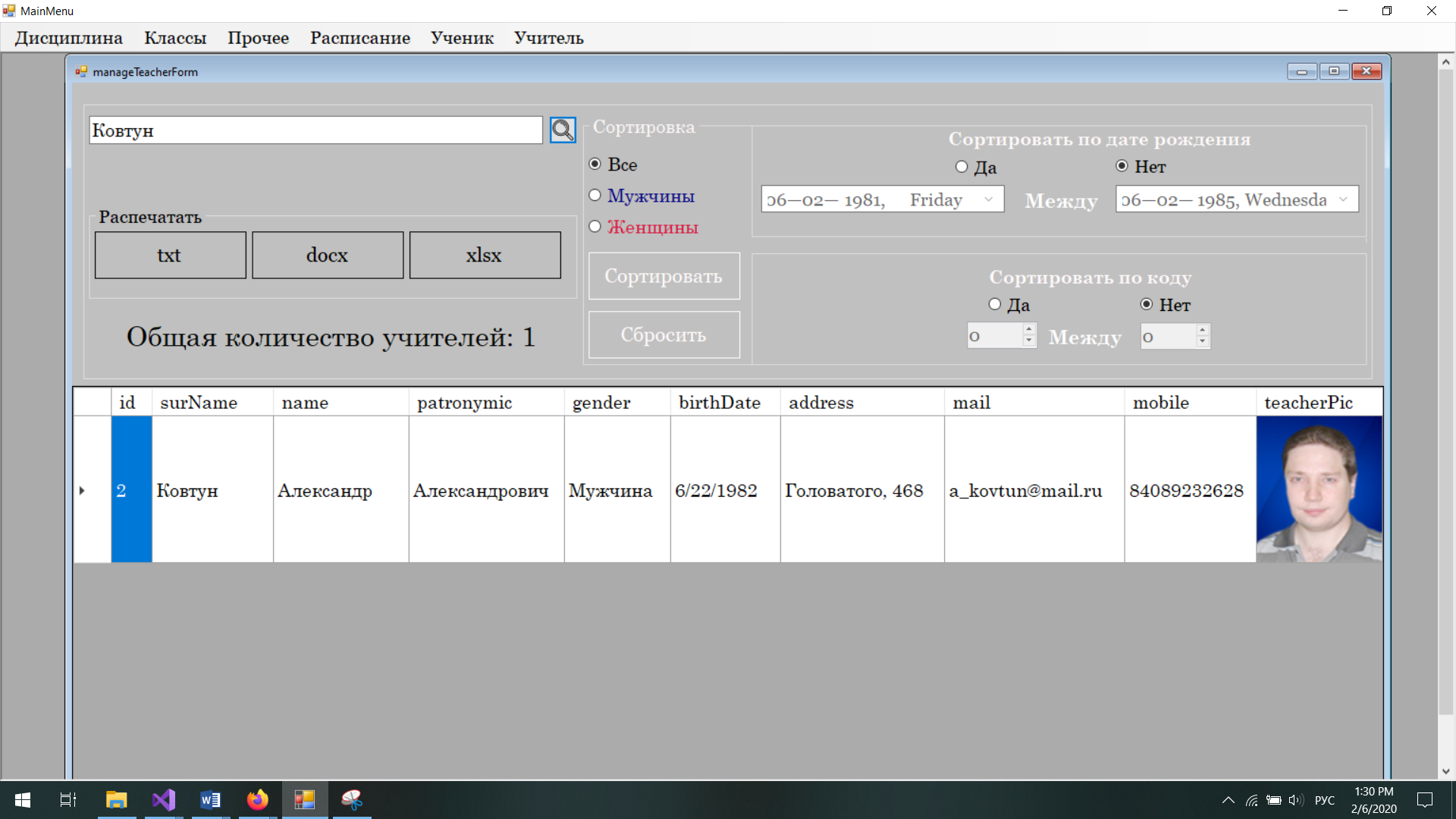


Рисунок 8 – Пример добавление учителя

Рисунок 9 – Пример изменения и удаление учителя

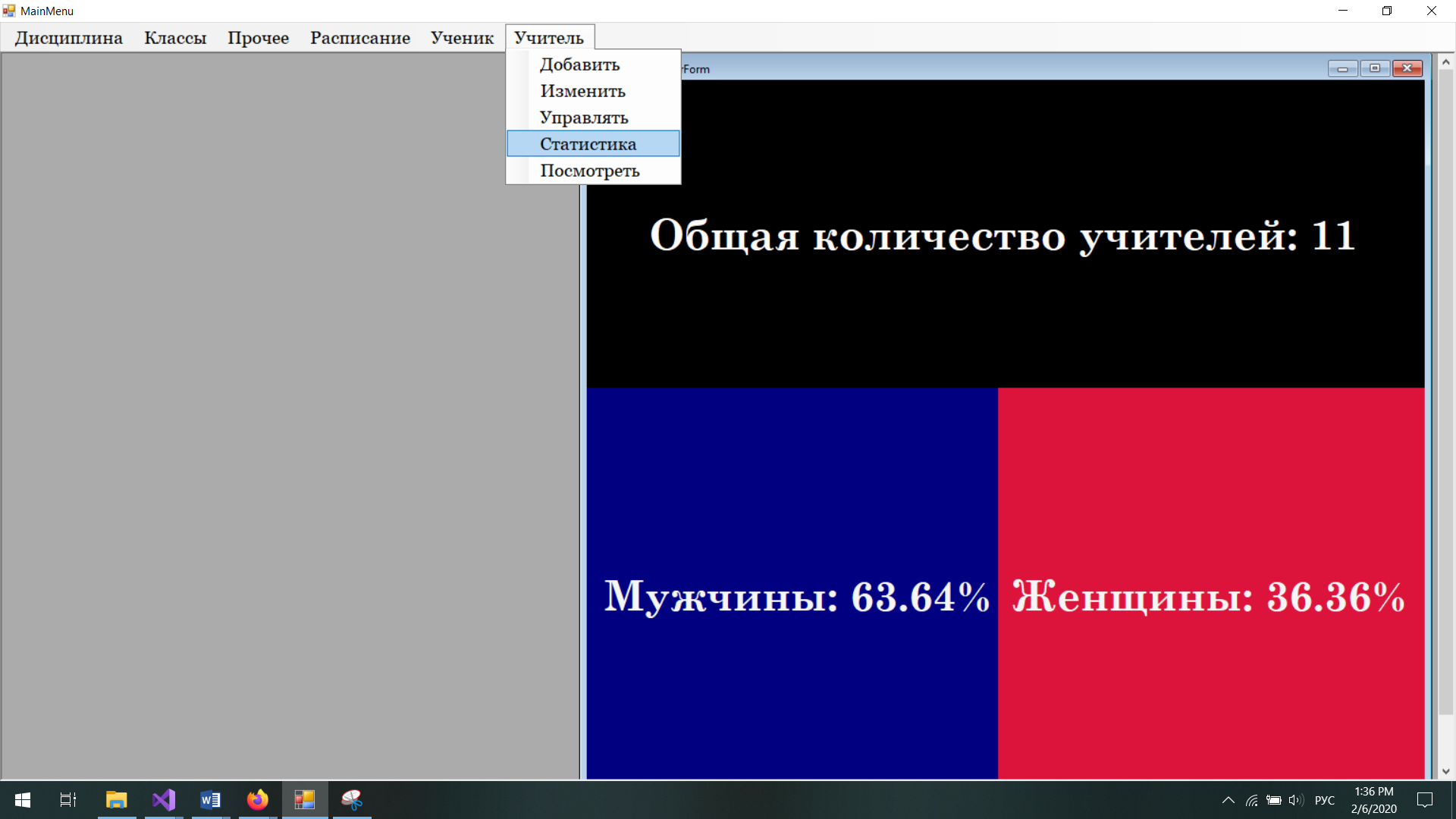
Рисунок 10 – Пример управление учителями

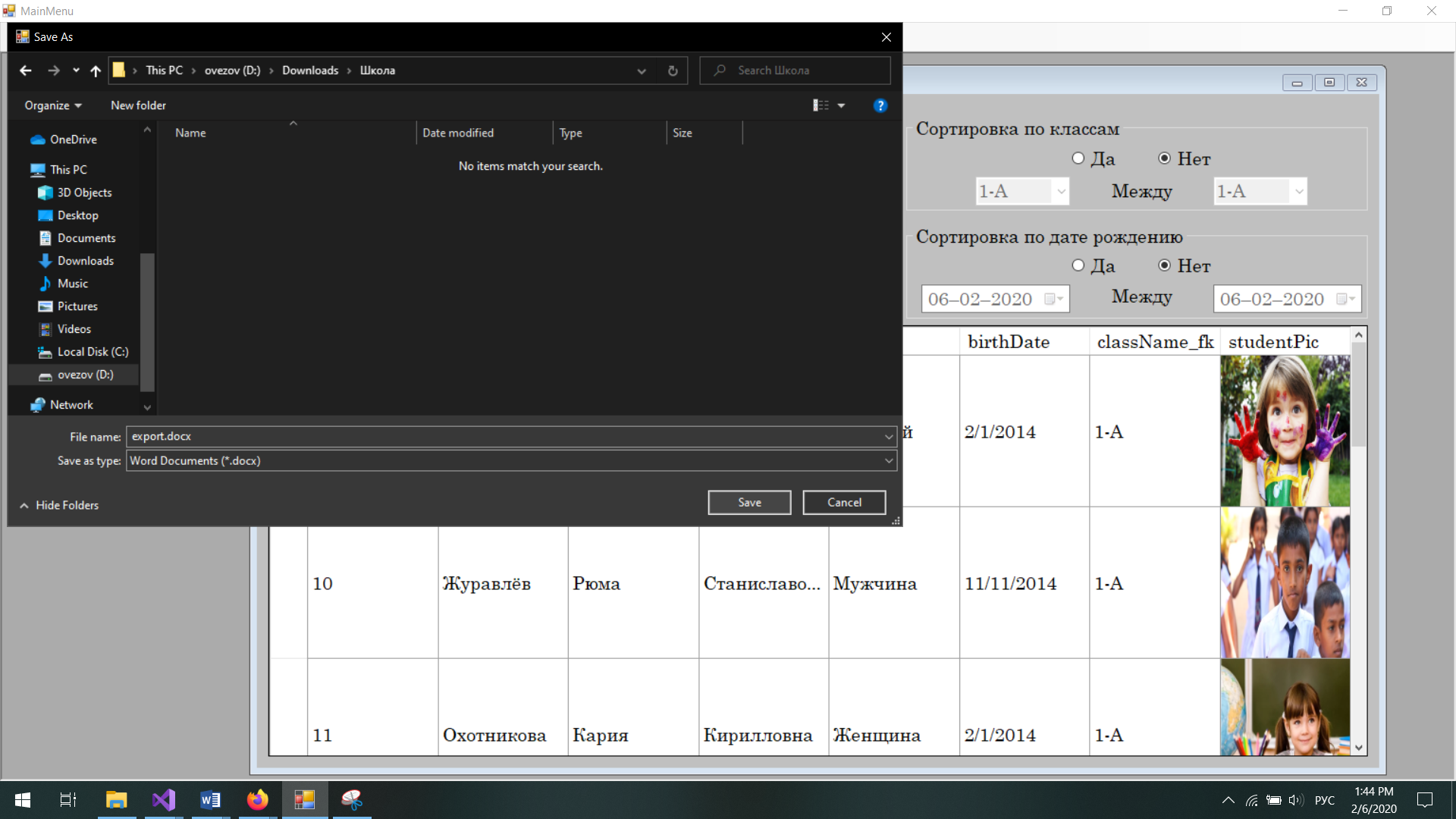
Рисунок 11 – Результат сортировки учителей

Рисунок 12 – Результат поиска учителя

Если пользователь захочет посмотреть статистику сколько учителей в данной школе работает и сколько из них мужчины или женщины можно увидеть в меню вкладке «статистика» (рис. 16).

Чтобы сохранить полученные данные пользователю необходимо выбрать место сохранении документа в персональном компьютере (рис. 17) после чего выполняется команда сохранение данных, например, в виде документа MS WORD (рис. 18).

Рисунок 13 – Статистика учителей

Рисунок 14 – Окно сохранении

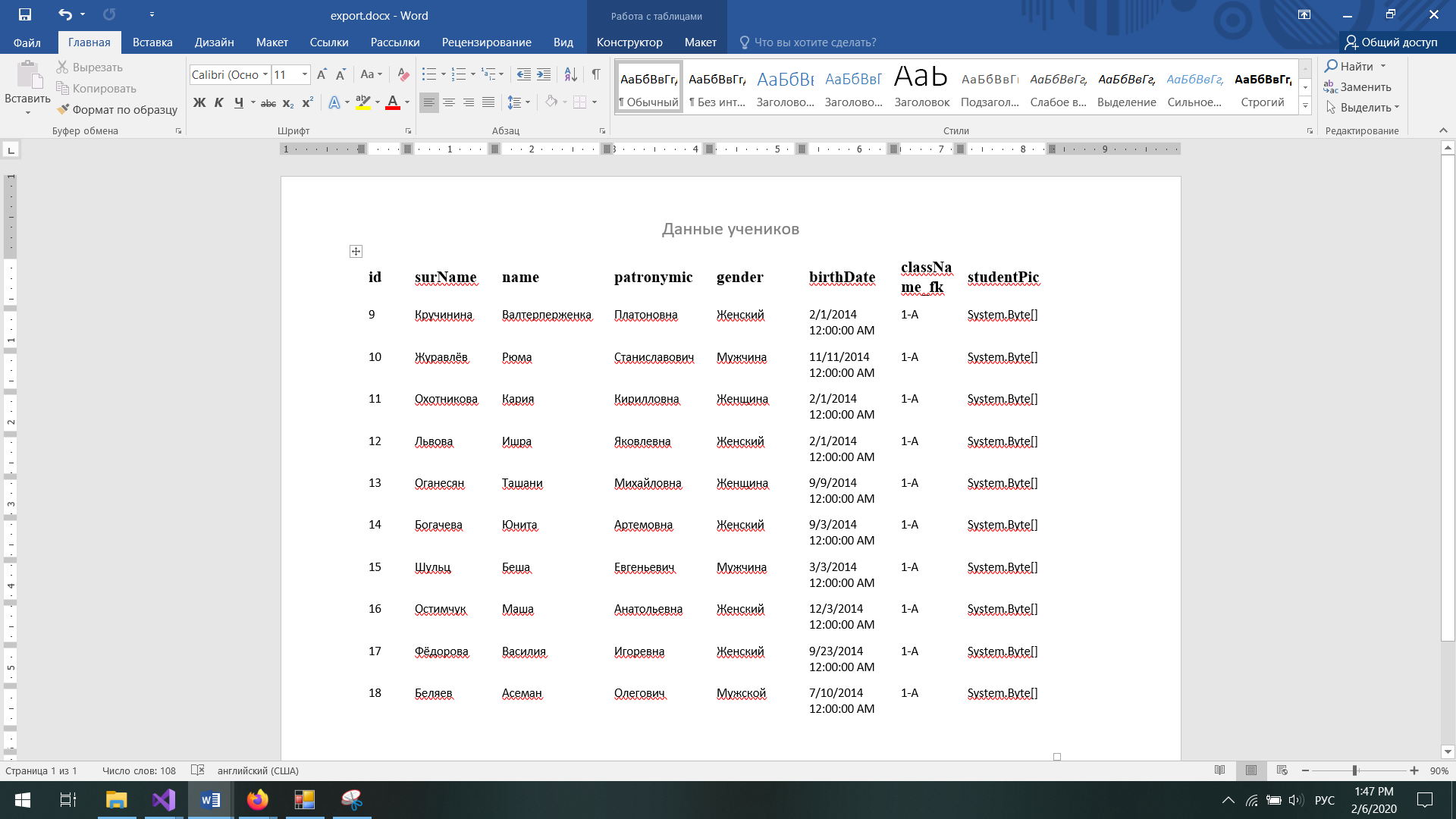


Рисунок 15 – Результат сохранении

# **Заключение**

В ходе проделанной курсовой работы было создано подробное описание процесса разработки программного обеспечения для системы учета вступительных экзаменов с использованием средств СУБД.

Были применены следующие средства разработки: MS Visual Studio, MY SQL Server, PHPMYADMIN.

В результате были смоделированы: концептуальная модель данных, ER-диаграмма, которая легла в основу приложения в части обработки данных. Также было смоделировано и само приложение для работы в данной автоматизированной системе.

Конечно программа требует улучшение в визуализации и в функциональности и будет еще лучше если расширять программу и создать полна функциональную школьную систему. Но сама же программа оказалась устойчивой за счёт обработки исключений, появляющихся вследствие попытки внести и сохранить данные, имеющие либо неверное введённые поля, либо имеющие точно такие же значения.

**Список использованных источников**

1. Голицына, О. Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, **2016**. - 400 c.

2. Владимир, Михайлович Илюшечкин Основы использования и проектирования баз данных / Владимир Михайлович Илюшечкин. - М.: Юрайт, **2018**. **- 516** c.

3. Архитектура и технологии IBM eServer zSeries / В.А. Варфоломеев и др. - М.: Интернет-университет информационных технологий, 2017. - 640 c.

4. Полис Г., Огастин Л., Мадхар Д. Разработка программных проектов: на основе Rational Unified Process (RUP). – М.: ООО «Бином- Пресс», 2019. – 352 с.

5. Nick Symmonds, Nicholas Symmonds, Data Entry and Validation with C# and VB. NET Windows Forms; **Гостехиздат** - Москва, **2017**. - **103** c.

# Приложение А

**Проверка курсового проекта на сайте antiplagiat.ru**

# Приложение Б