# 

# ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc40026496)

[1 ОСНОВНАЯ ЧАСТЬ 7](#_Toc40026497)

[1.1 Назначение и цели создания системы 7](#_Toc40026498)

[1.2 Обзор и анализ предметной области 7](#_Toc40026499)

[1.3 Жизненный цикл базы данных 8](#_Toc40026500)

[1.4 Выбор и характеристика СУБД 9](#_Toc40026501)

[1.5 Выбор и характеристика среды разработки 9](#_Toc40026502)

[1.6 Выбор и характеристика языка программирования 10](#_Toc40026503)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 11](#_Toc40026504)

[2.1 Постановка задачи 11](#_Toc40026505)

[2.2 Архитектура информационной системы 11](#_Toc40026506)

[2.2.1 Основные этапы проектирования приложения 11](#_Toc40026507)

[2.2.2 Архитектура клиент-сервер 12](#_Toc40026509)

[2.2.3 Навигационная схема приложения 13](#_Toc40026510)

[2.2.4 Архитектура классов приложения 13](#_Toc40026511)

[2.3 Логическая модель базы данных 15](#_Toc40026512)

[2.4 Нормализация таблиц 14](#_Toc40026513)

[2.5 Описание таблиц 15](#_Toc40026514)

[2.6 Разработка приложения 19](#_Toc40026515)

[2.6.1 Диаграмма вариантов использования 19](#_Toc40026516)

[2.6.2 Решение главной задачи проекта 19](#_Toc40026517)

[2.6.3 Защита информационной системы ……………………………..](#_Toc40026519)20

[2.6. Скриншоты интерфейса 21](#_Toc40026520)

[2.7 Инструкция пользователю 26](#_Toc40026521)

[2.7.1 Общие сведения об информационной системе 26](#_Toc40026522)

[2.7.2 Требования к техническим средствам 226](#_Toc40026523)

[2.7.3 Требования к программным средствам 27](#_Toc40026524)

[2.7.4 Настройка информационной системы 27](#_Toc40026525)

[2.7.5 Формы ввода 27](#_Toc40026526)

[2.7.6 Отчеты 27](#_Toc40026527)

[ЗАКЛЮЧЕНИЕ 27](#_Toc40026528)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 29](#_Toc40026529)

[ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных, на создание учетных записей пользователей, на назначение привилегий 29](#_Toc40026530)

[ПРИЛОЖЕНИЕ Б Код программы 54](#_Toc40026531)

# ВВЕДЕНИЕ

Темой проекта является разработка информационной системы для

колледжа. В современном учебном заведении без локальной базы данных с учетом всех необходимых данных для поддержания информационной системы обойтись невозможно. Многие школы, колледжи и другие учебные заведения используют онлайн сервисы для хранения данных, так же обычно используются приложения Access или Exсel, для тех же нужд. Так или иначе, в современном мире невозможно обойтись без серьезной информационной системы. Для этих целей многие предприятия используют базы данных написанные на языке программирования SQL, но этого недостаточно для полноценной удобной информационной системы. Для удобства пользования конечным пользователем требуется приложение, которое будет содержать и управлять базой данных.

Отталкиваясь от этой идеи мне стало интересно разработать приложение, которое позволит осуществлять полное управление над базой данных колледжа включая: добавление, редактирование, поиск и удаление различных элементов, связанных с моим учебным заведением.

Основные задачи, которые необходимо выполнить для того, чтобы разработать данное приложение:

* подробное изучение функциональных возможностей WPF.Net Framework,
* освоение тонкостей проектирования баз данных,
* изучение Entity Framework,
* разработка функций для работы с базой данных,
* создание функций для работы с данными внутри приложения,
* проектирование интерфейса программы,
* написание функций для корректного сохранения и обновления данных,
* создание функций для корректного отображения содержимого страниц,
* проектирование архитектуры и логики приложения.

Используемые в разработке паттерны и методы:

* объектно-ориентированное программирование,
* моделирование.

Обоснование структуры курсового проекта:

1. введение: в данном разделе описываются цели и задачи проекта, а также его актуальность, так как изначально необходимо определить, что предстоит изучать и делать в проекте,
2. основная часть: этот раздел включает в себя анализ предметной области, для которой разрабатывается данное приложение, назначения и цели разработки, а также жизненный цикл базы данных,
3. специальная часть: в данном разделе описаны архитектурные решения, которые применяются в разработке, а также процесс разработки программного обеспечения и базы данных,
4. заключение: вывод о проделанной работе и полученных навыках за время разработки приложения,
5. список используемых источников: этот раздел содержит в себе перечень ресурсов, которые были использованы для получения необходимой информации во время разработки.

# ОСНОВНАЯ ЧАСТЬ

Назначение и цели создания системы

Приложение создается для разработки информационной системы

колледжа. База данных приложения содержит информацию о студентах, группах, преподавателях, предметах, специальностях, отделениях и экзаменах.

Цель: создать приложение средствами которого можно искать, добавлять и редактировать базу данных колледжа, со всеми вытекающими последствиями

Назначение системы: обеспечить доступ к редактированию базы данных колледжа.

Базовые сущности этой предметной области: студенты, группы, преподаватели, экзамены, предметы.

В системе должны существовать два вида пользователей: администратор и пользователь.

Обзор и анализ предметной области

Предметной областью разработки является база данных колледжа. В этой отрасли существует обширный набор вводных параметров. В области некоторых параметров есть данные о студентах, их личные данные, именах преподавателей и номерах групп. У каждого студента есть номер, по которому он приписывается в ту или иную группу и получает ту или иную оценку за экзамен. У каждого экзамена есть свой учитель, который его принимал и студент, который его сдавал.

Список основных сущностей:

* студенты – это ученики колледжа. Атрибуты студентов: номер, фамилия, имя, отчество, дата рождения, номер телефона, серия и номер паспорта, адрес,
* преподаватели – это учителя, имеющие право принимать экзамены и вести определенные предметы. Атрибуты преподавателей такие же, как атрибуты студентов, но включают в себя стаж и опыт работы
* предметы – конкретные дисциплины, по которым проводятся экзамены и в области которых учителя могу ставить допуски и оценки. Атрибуты предметов: идентификатор, название, специальность, количество учебных часов.
* экзамены – оценка студента за предмет. Атрибуты экзаменов: идентификатор, допуск, оценка, дата сдачи, идентификатор учителя, идентификатор предмета.
* группы – группа к которой прикреплен студент. Атрибуты групп: идентификатор, номер группы, учебный год, отделение, куратор, специальность.
* Специальности, к которым относятся студенты и предметы. Атрибуты специальностей: идентификатор, название, учебный год, тип, номер группы,
* Отделения – место, отвечающие за студентов, группы и имеющие своих глав. Атрибуты отделений: идентификатор, номер, учебный год, глава отделения, номер группы, идентификатор главы отделения.

Жизненный цикл базы данных

Жизненный цикл – это процесс проектирования, реализации и управления базой данных. Он состоит из:

1. Анализа предметной области, который включает в себя планирование разработки (15.02.2020 – 25.02.2020), определение требований (15.02.2020 – 25.02.2020), сбор и анализ требований (15.02.2020 – 25.02.2020),
2. Проектирования базы данных, которое состоит из создания концептуальной модели (26.02.2020), логической модели (26.02.2020), физической модели (27.02.2020),
3. Реализации, в процессе которой пишутся скрипты на создание базы данных, связей сущностей в базе данных, а также разрабатывается интерфейс приложения и его функции (01.03.2020 – 05.05.2020).

Выбор и характеристика СУБД

Для разработки базы данных была выбрана среда Microsoft SQL Server Management Studio 2018, так как она является удобной для использования и не требует долгого освоения тонкостей для создания полноценной базы данных под программу, а также с ней можно работать, используя Entity Framework. Эта СУБД использует язык SQL (T-SQL) для написания необходимых сущностей и заполнения их данными.

При сравнении SQL Server с MySQL можно выделить несколько главных различий:

* первый предназначен для работы с платформой .Net, а второй для работы с PHP,
* синтаксис,
* данные SQL Server хранятся в единой системе Microsoft, в то время как данные MySQL хранятся на движках.

Весомыми плюсами SQL Server являются простота понимания принципов работы, синтаксиса, а также возможность создавать объекты баз данных либо с помощью скрипта, либо вручную, с помощью конструктора.

Выбор и характеристика среды разработки

Для разработки приложения была выбрана среда программирования Microsoft Visual Studio 2019, потому что эта среда разработки обеспечивает возможность создания приложений с удобным интерфейсом с помощью языка C# и системы построения клиентских приложений WPF.Net Framework.

Microsoft Visual Studio 2019 – продукт компании Microsoft, включающий в себя интегрированную среду разработки программного обеспечения, а также ряд других инструментальных средств.

Некоторые преимущества:

* удобная отладка и поиск ошибок,
* возможность создания специальной библиотеки для тестирования,
* доступность NuGet пакетов, необходимых для работы приложения,
* подробная документация об ошибках, располагающаяся на сайте компании Microsoft,
* интеллектуальность системы, предупреждающая об ошибках и помогающая упростить и улучшить код.

Данную среду разработки можно сравнить с Visual Studio Code. Она является менее удобной, так как не представляет такого обширного количества средств для отладки и тестирования.

Выбор и характеристика языка программирования

Для написания кода приложения был выбран язык C#, так как он является предпочтительным для меня, а также с целью повышения навыков его использования в различных целях.

C# – объектно-ориентированный язык программирования от компании Microsoft, предназначенный для разработки приложений для платформы Microsoft.Net.

Некоторые преимущества:

* поддерживается ООП,
* удобный синтаксис,

В сравнении с языком Visual Basic, C# является более удобным и лаконичным, так как был изначально спроектирован под платформу .Net.

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Постановка задачи

Главная задача проекта – спроектировать средствами MS SQL SERVER базу данных колледжа и создать приложение средствами Microsoft Visual Studio 2019 и Microsoft SQL Server Management Studio 18, позволяющее осуществлять поиск, редактирование, просмотр, добавление и удаление атрибутов базы данных для администратора, просматривать, редактировать некоторые данные для пользователя.

Предусмотреть чтобы:

* единовременно один студент мог находится лишь в одной группе, на одной специальности
* единовременно в одной специальности могли находится множество групп,
* единовременно один экзамен по определенному предмету могли сдавать все студенты,
* администратор мог добавлять, искать, редактировать и удалять информацию о студентах, преподавателях, правах доступа, экзаменах, группах, отделениях, специальностях и предметах,
* пользователи могли просматривать, искать и редактировать доступную им информацию.

В системе должны существовать два вида пользователей: администратор и пользователь.

Администратор может выполнять следующие действия:

* добавлять, искать, редактировать и удалять всю доступную информацию и наделять правами доступа администратора пользователей,
* может создавать рейсы, аэропорты, типы самолётов и самолёты.

Пользователь может выполнять следующие действия:

* Просматривать, редактировать и искать доступную ему информацию.

2.2 Архитектура информационной системы

2.2.1 Основные этапы проектирования приложения

Первым пунктом разработки являлось создание первичной версии базы данных для последующей доработки.

После успешного завершения проектирования базы данных нужно было приступать к разработке окна и страниц приложения.

Первой была создана страница авторизации с возможностью перехода к регистрации или инструкции по эксплуатации приложения.

После этого было разработано меню программы, позволяющее авторизованным пользователям просматривать главное меню приложения, в котором хранится главная информация, в соответствии с их правами доступа.

На последнем этапе были созданы различия между правами доступа у пользователей. Следующим шагом была доработка форм ввода данных с целью защиты от неверного ввода информации.

2.2.2 Архитектура клиент-сервер

Клиент-сервер – архитектура, в которой задачи или сетевая нагрузка распределены между поставщиками услуг, которые называются серверами, и заказчиками услуг – клиентами. Серверы ожидают запрос от клиента и передают им свои ресурсы в виде данных. Модель данной архитектуры легко представить графически (Рисунок 2).

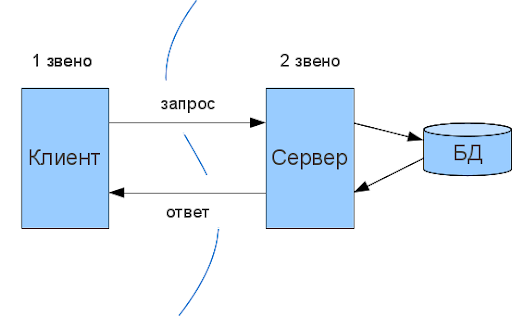


Рисунок 1 - модель архитектуры клиент-сервер.

2.2.3 Навигационная схема приложения

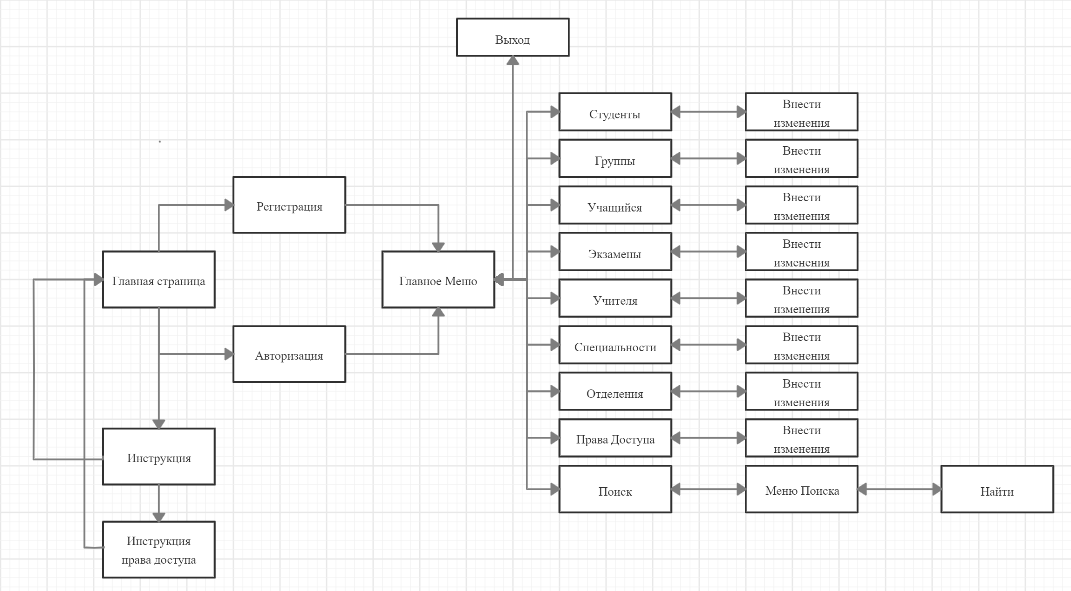


Рисунок 2 – Навигационная схема приложения

**2.2.4 Архитектура классов приложения**

Для создания оптимальной архитектуры в среде Microsoft Visual Studio 2019 я создал библиотеку классов Interface отвечающую за интерфейс приложения и его визуальные функции, такие как:

* создание страниц,
* корректное отображение элементов на странице,
* навигация,
* сохранение экземпляров страниц и других необходимых элементов с целью предотвращения создания дубликатов, а также обеспечения быстрого доступа,
* обработка событий, вызванных пользователем,
* проверка вводимых пользователем данных,
* отображение полученных данных их базы.

Для основных работ с базой данных, я пользовался встроенными функциями Entity Framework в Visual Studio 2019 для языка C#, которыми добавил нужную базу данных DBStudents, а так-же:

* соединение с базой данных,
* получение данных из базы,
* добавление данных в базу,
* изменение данных в базе,
* соединение данных таблиц,
* формирование экземпляров, с которыми могут корректно работать классы Interface
* формирование экземпляров, с которыми может корректно работать база данных,
* проверки существования записей с указанными параметрами.

Классы, содержащиеся в Interface:

* MF, SE, Roli.

Таблицы, содержащиеся в DBStudents:

* Department1,
* Exam1,
* Group,
* Rols,
* Specialty,
* Student,
* Studying,
* Teacher,
* Subject,
* TS,
* SubSpec,
* Sysdiagram.

Так же для создания приложения я использовал страницы:

* Pravainstr.xaml,
* Instr.xaml,
* MainMenu1.xaml,
* Page1.xaml,
* Page2.xaml,

2.3 Нормализация таблиц

Первая нормальная форма – все атрибуты атомарны.

Вторая нормальная форма – выполнены условия первой нормальной формы, а также каждый неключевой атрибут неприводимо зависит от первичного ключа. Неприводимость – отсутствие атрибутов, от которых можно вывести функциональную зависимость.

Пример сущностей во второй нормальной форме (Рисунок 6).

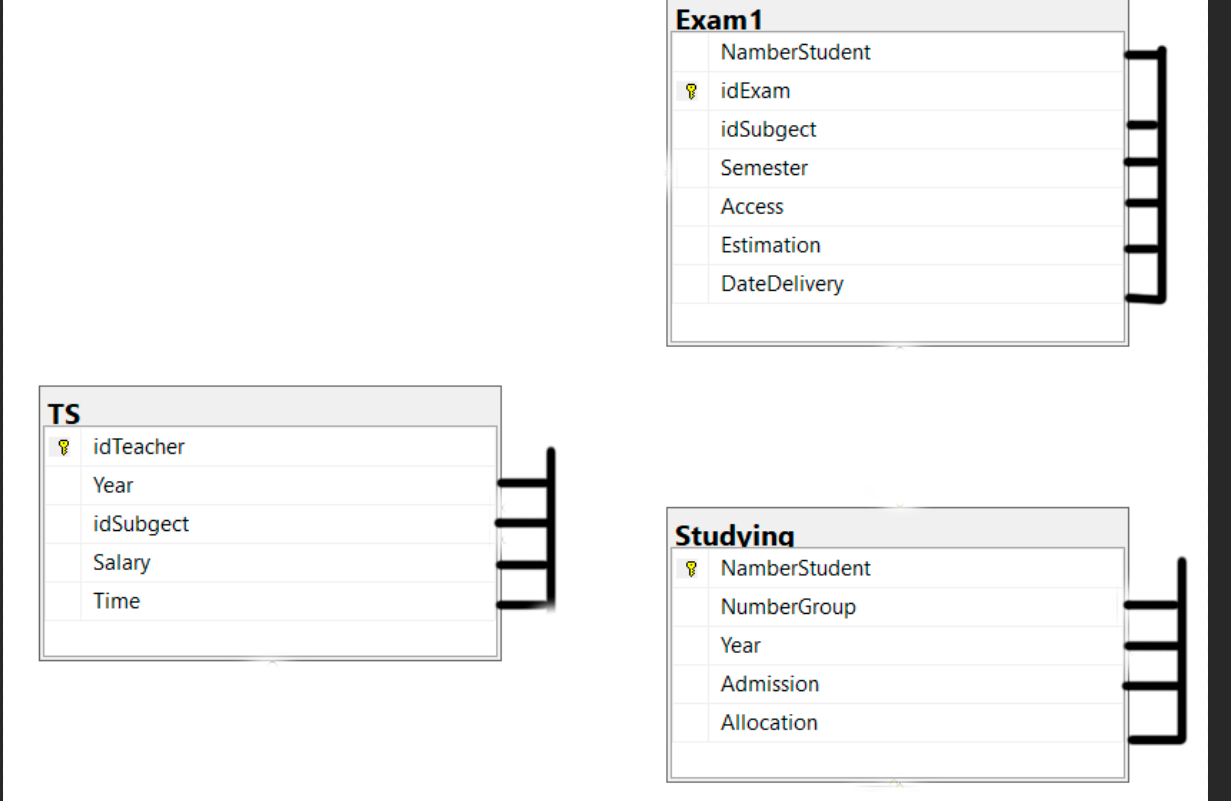


Рисунок 3 – пример второй нормальной формы

Третья нормальная форма – соблюдены условия второй нормальной формы, а также каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

База данных приведена к третьей нормальной форме.

2.3 Логическая модель базы данных

Логическая модель базы данных, используемой в приложении (Рисунок 5).

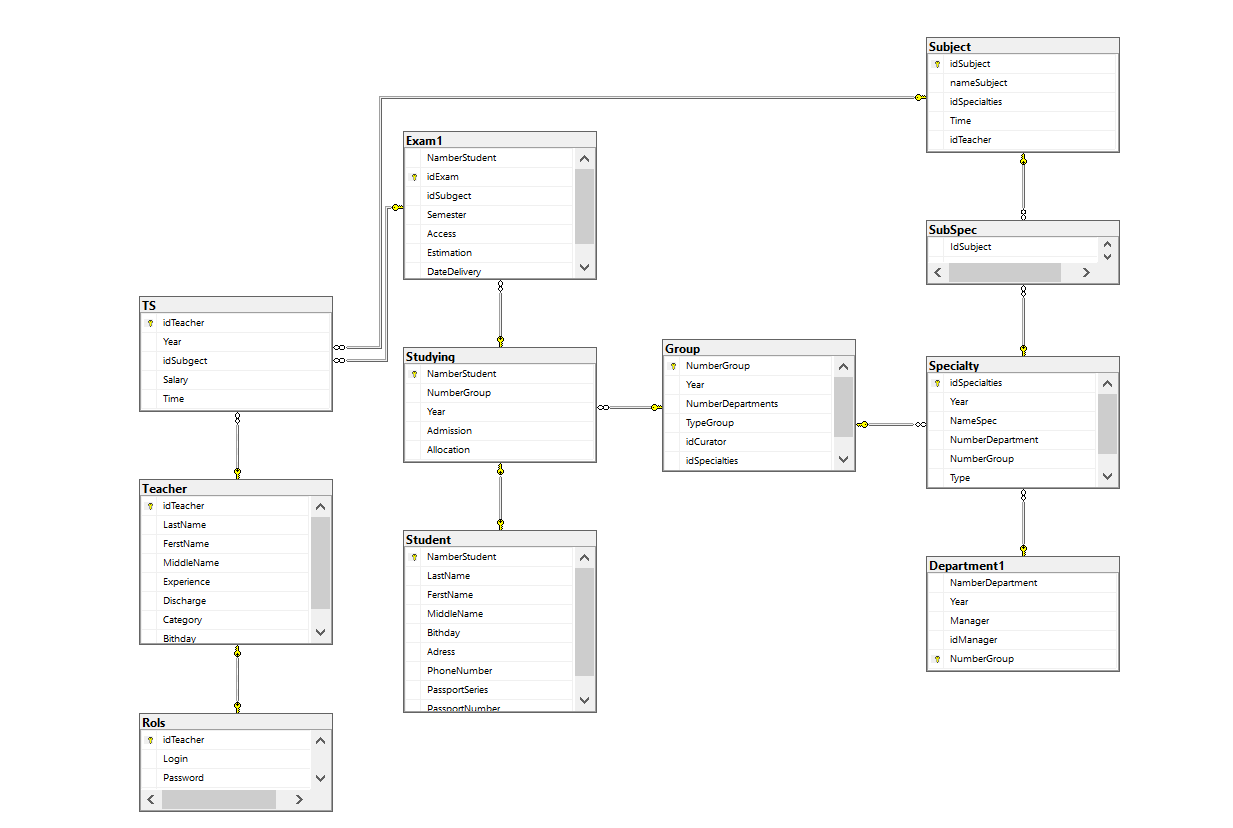


Рисунок 4 – Логическая модель базы данных

2.4 Описание таблиц

В этом разделе показаны названия таблиц, их атрибуты, свойства и типы атрибутов, первичные и внешние ключи.

Таблица студентов хранит основные данные по студентам колледжа (Рисунок 5)

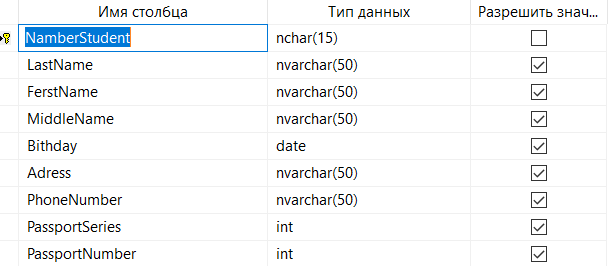


Рисунок 5 – Таблица Student

Таблица Группы, хранит основные данные по группам колледжа (Рисунок 6)

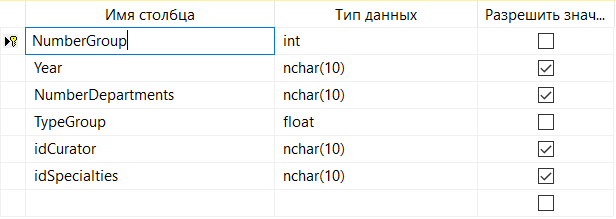


Рисунок 6 – Таблица Group

Таблица Роли хранит основные данные по ролям (Рисунок 7), в приложении они называются “Права доступа” и при необходимости задаются Администратором

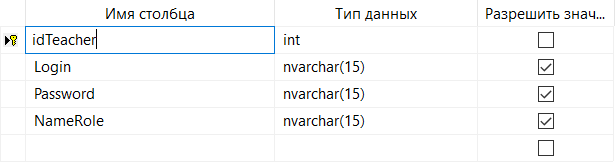


Рисунок 7 – Таблица Rols

Таблица Учащийся позволяет связать студента и группу, чтоб в 1 определенный студент мог учиться в одной определенной группе (Рисунок 8)

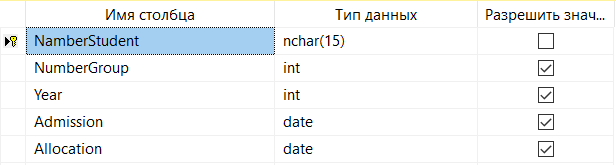


Рисунок 8 – Таблица Studying

Таблица Предмет хранит информацию о предметах в колледже (Рисунок 9)

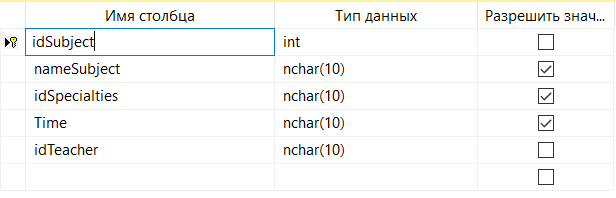


Рисунок 9 – Таблица Subject

Таблица Учителя хранит информацию о учителях колледжа (Рисунок 10)

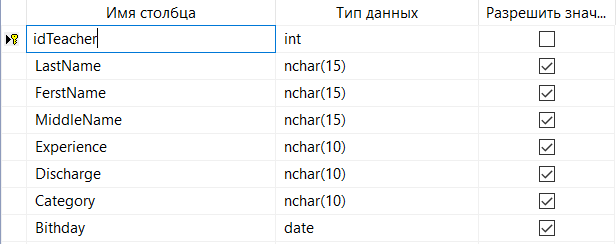


Рисунок 10 – Таблица Teachers

Таблица Специальности хранит информацию о специальностях колледжа (Рисунок 11)

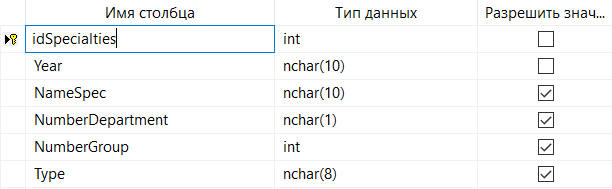


Рисунок 11 – Таблица Specialty

Таблица Отделения хранит информацию о отделениях колледжа (Рисунок 12)

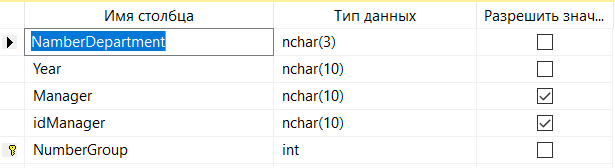


Рисунок 12 – Таблица Departament1

Таблица Экзамены хранит информацию о существующих в колледже экзаменах (Рисунок 13)

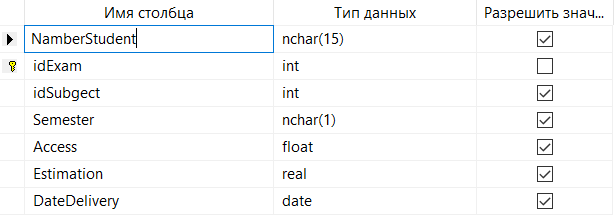


Рисунок 13 – Таблица Exam1

Таблица Предмет-Специальность соединяет таблицы Предметы и Специальности (Рисунок 14)

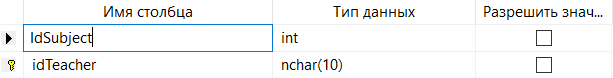


Рисунок 14 – Таблица SubSpec

Таблица Учителя-Предметы соединяет таблицы Учителя и Предметы (Рисунок 15)

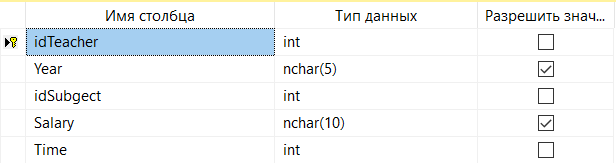


Рисунок 15 – Таблица TS

2.5 Разработка приложения

2.5.1 Диаграмма вариантов использования



Рисунок 16 – диаграмма вариантов использования

2.5.2 Решение главной задачи проекта

Главная задача проекта – управление базой данных студентов колледжа пользователем исходя из его прав доступа

Необходимо:

* определить роль текущего пользователя,
* в соответствии с этой ролью отобразить список доступной для просмотра и редактирования информации
* при выборе информации, которую нужно отредактировать, заменить ее на желаемую и нажать клавишу “Enter”, после чего нажать кнопку “Сохранить”,
* при выборе интересующей таблицы и нажатии кнопки “Добавить” создать новую строку для добавления соответствующей информации,
* осуществить возможность поиска по фамилии студента с возможностью распечатать полученную сводку данных по студенту,
* Осуществить возможность удаления информации частями, через возможности редактирования или же всей строки с данными при нажатии клавиши “Delete”.

2.5.3 Защита информационной системы

В разрабатываемом приложении предусмотрена защита всех форм ввода от неправильной и некорректной информации. При попытке сохранения неверных данных пользователю будет выведено сообщение об ошибке или поле ввода будет подсвечиваться красным цветом до исправления ошибки. При вводе корректных данных все будет успешно сохранено.

2.5.4 Скриншоты интерфейса

Все скриншоты сделаны с аккаунта администратора. В начале стартовой страницы (Рисунок 17) доступен выбор авторизоваться или зарегистрироваться, так же присутствует кнопка эксплуатации по применению

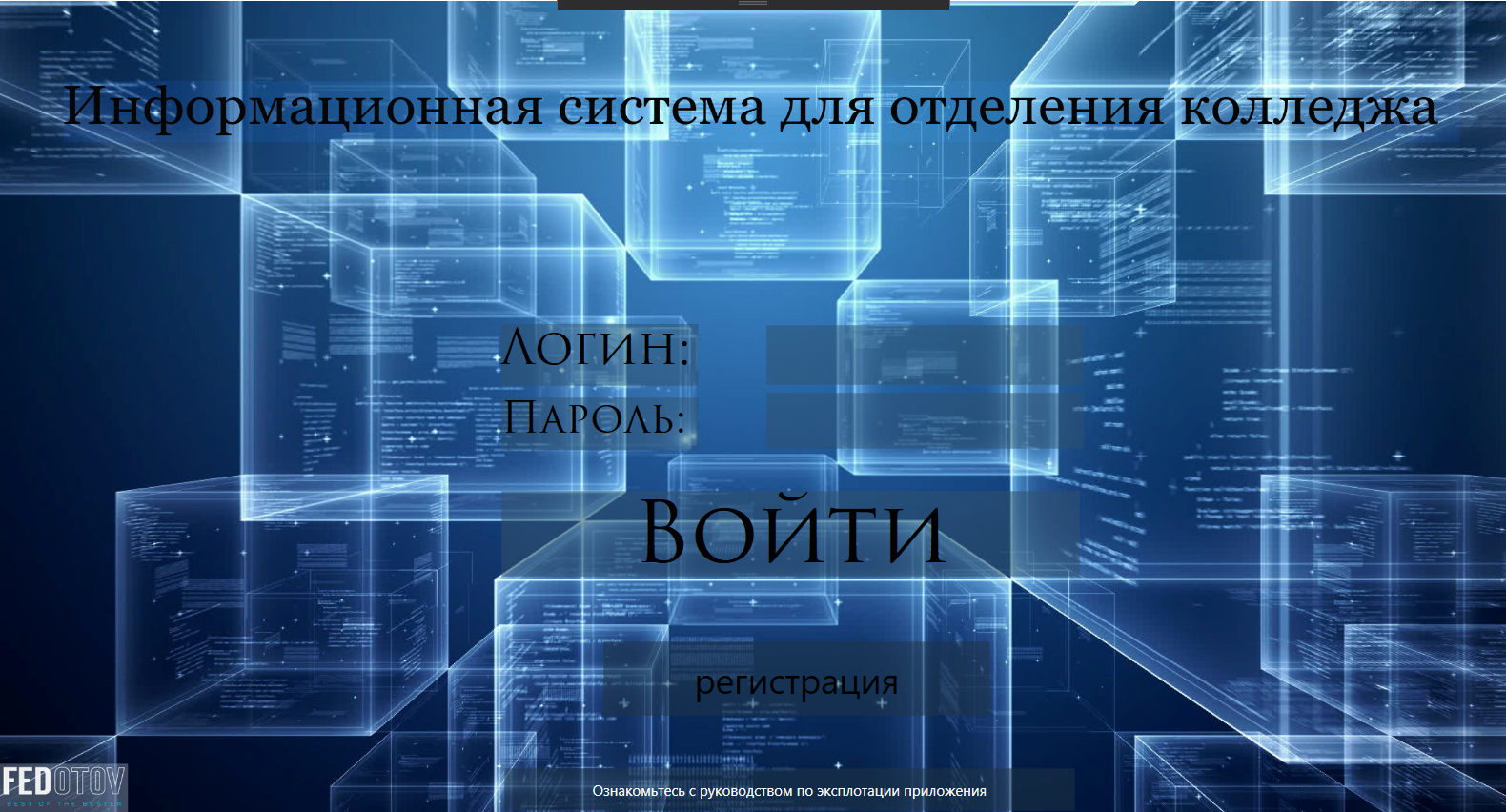


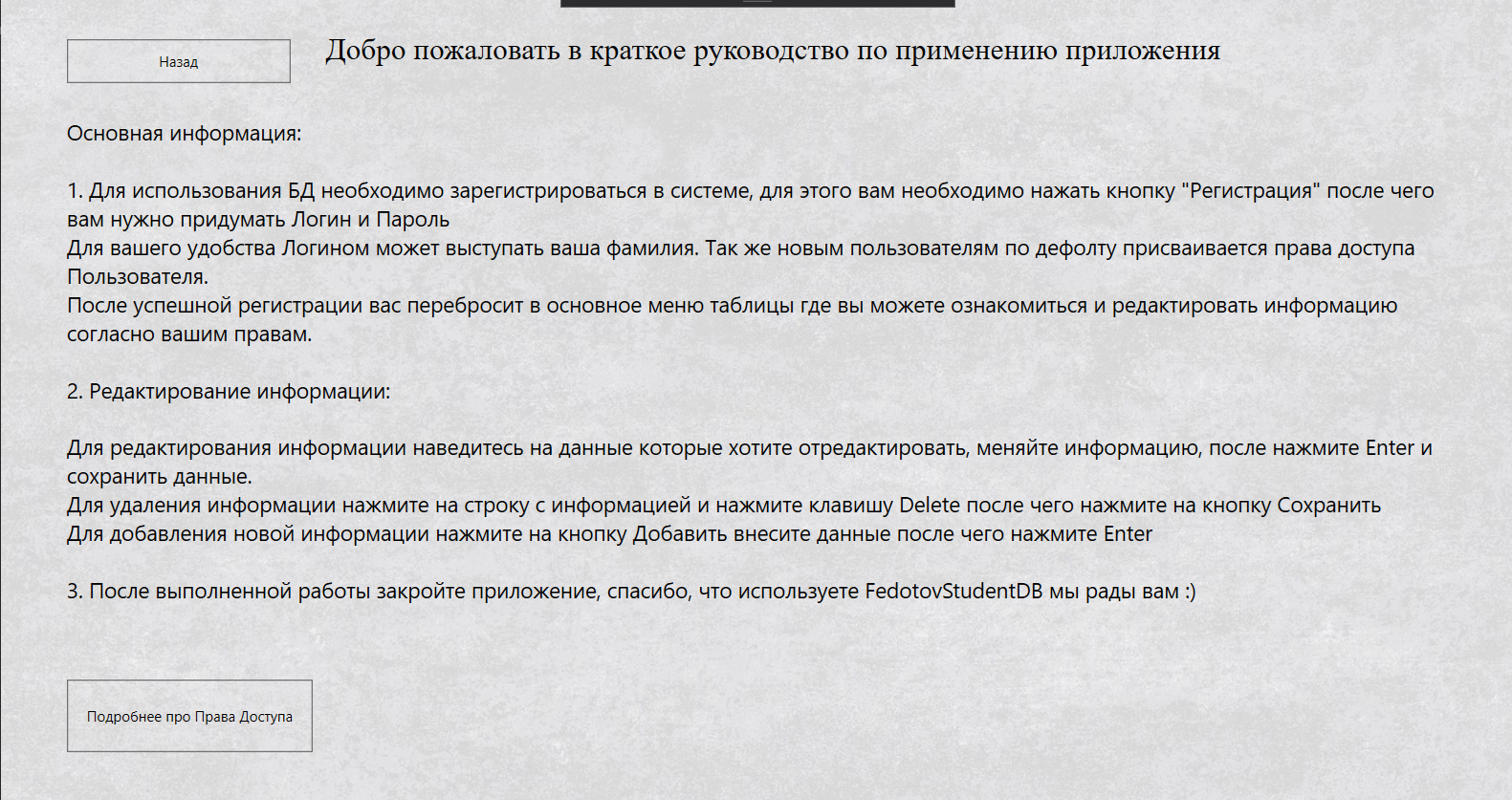
Рисунок 17 – стартовая страница авторизация 

Рисунок 18 – инструкция по эксплуатации приложения

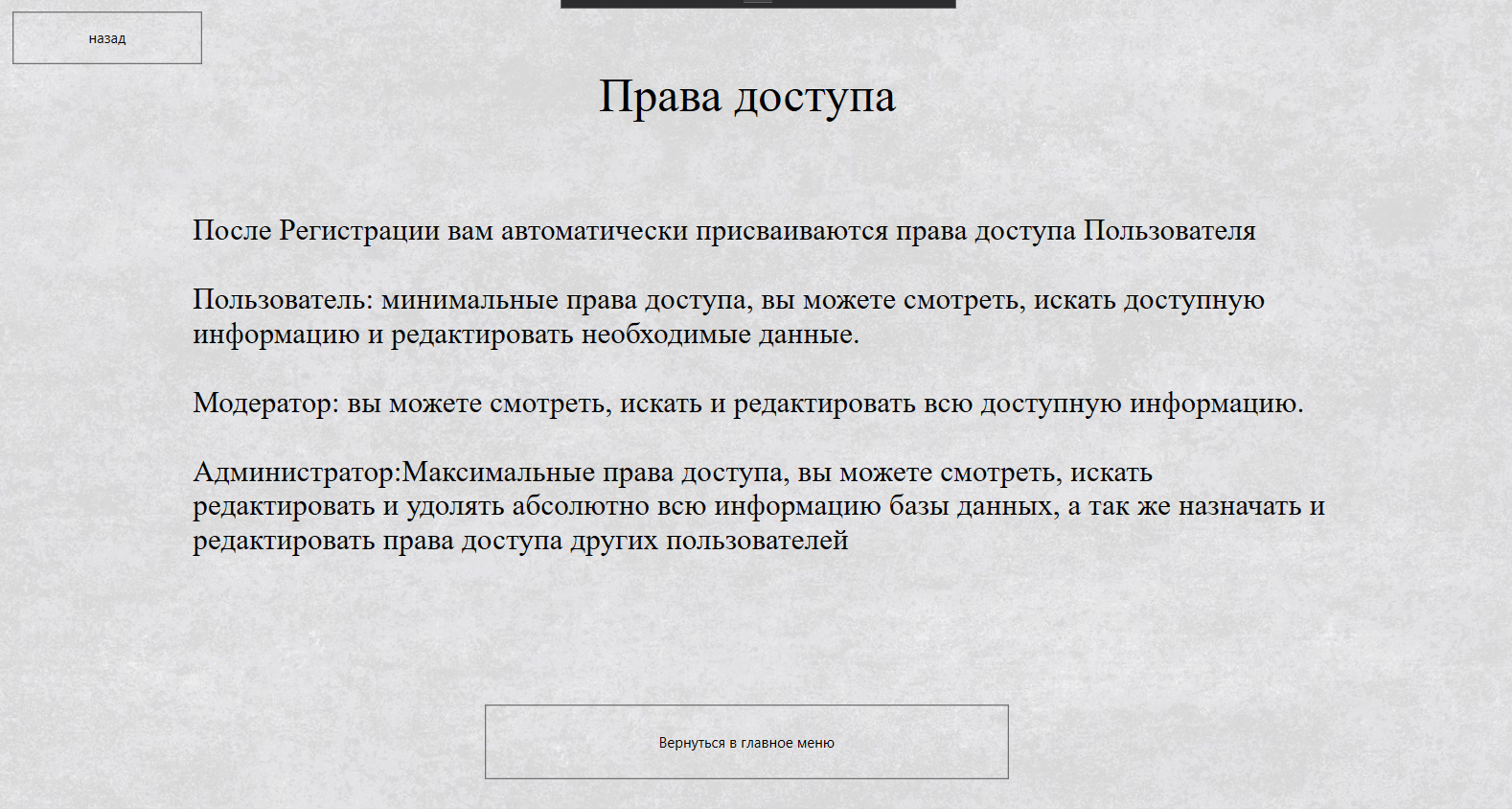


Рисунок 19 – сведения по правам доступа

После ознакомления с правами доступа незарегистрированный пользователь должен пройти регистрацию (Рисунок 20), после чего ему автоматически будут присвоены права доступа Пользователь

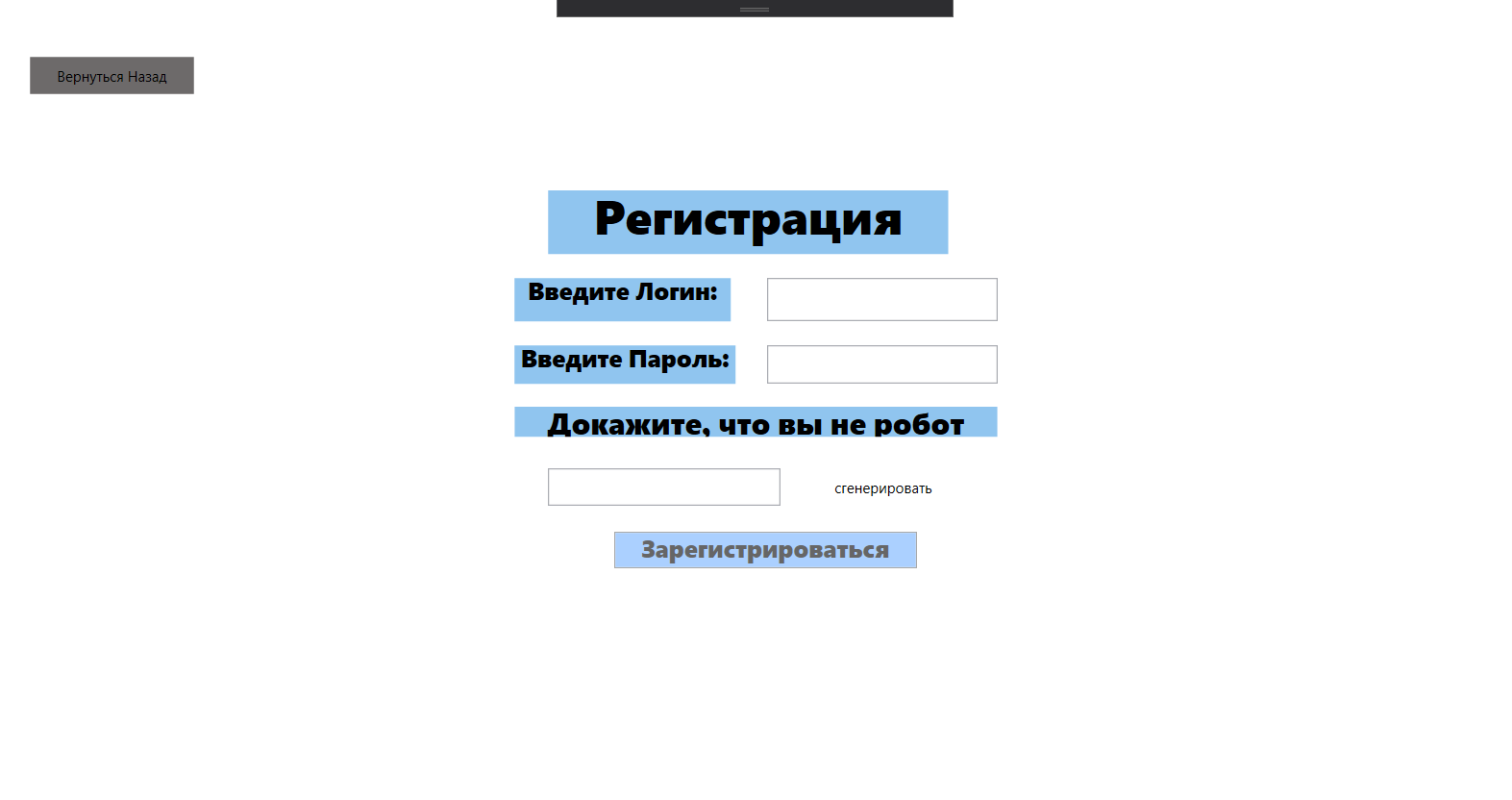


Рисунок 20 – регистрация

После авторизации или регистрации пользователь попадает в главное меню (Рисунок 21). У администратора и пользователя главные меню отличаются.



Рисунок 21 – главное меню администратора



Рисунок 22 – пример открытой таблицы

При нажатии на кнопку “Найдется все” открывается меню поиска (Рисунок 23)

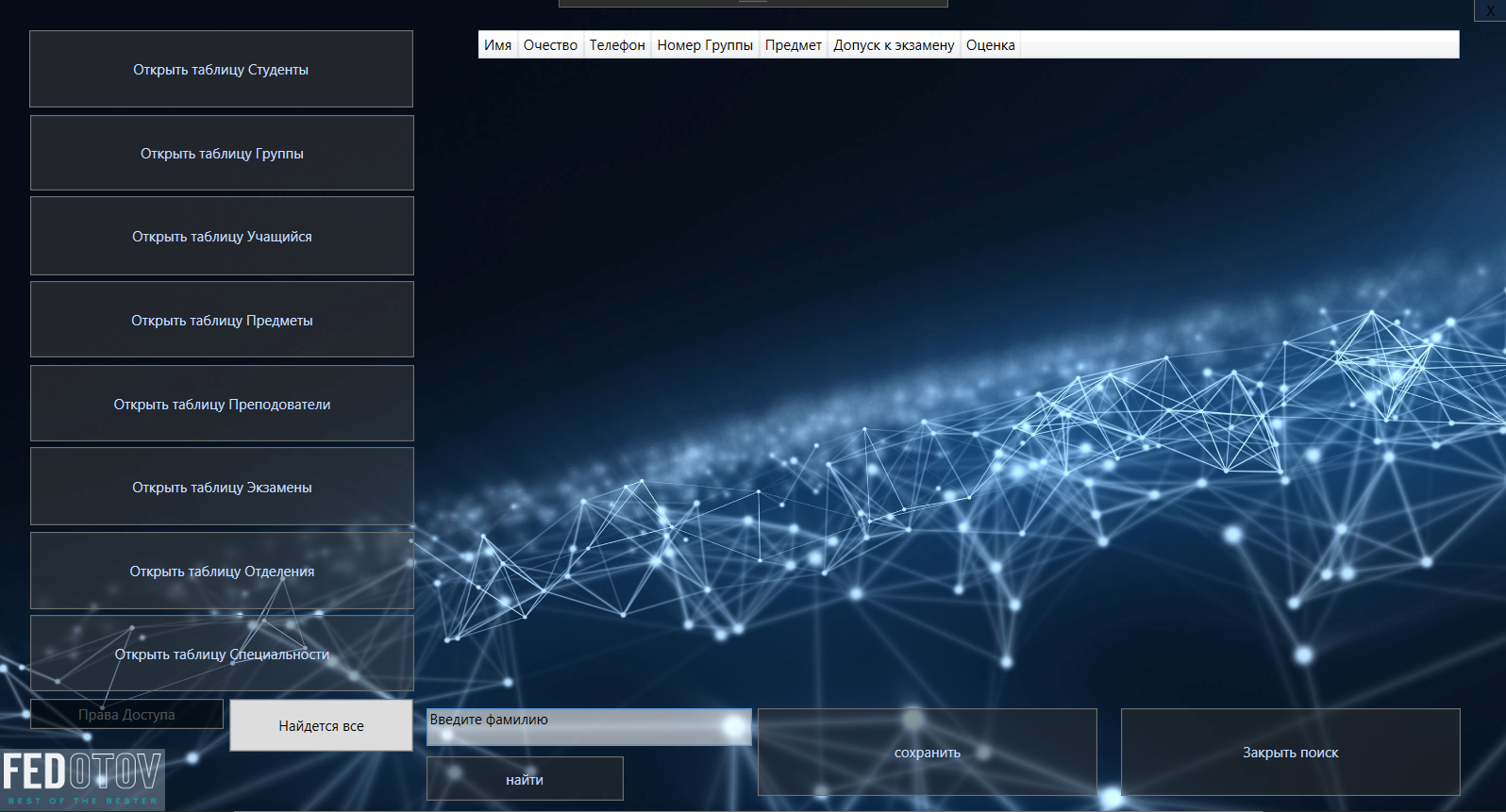


Рисунок 23 – меню поиска

После введения фамилии студента и нажатия кнопки “Найти” выводится вся доступная информация на студента (Рисунок 24)

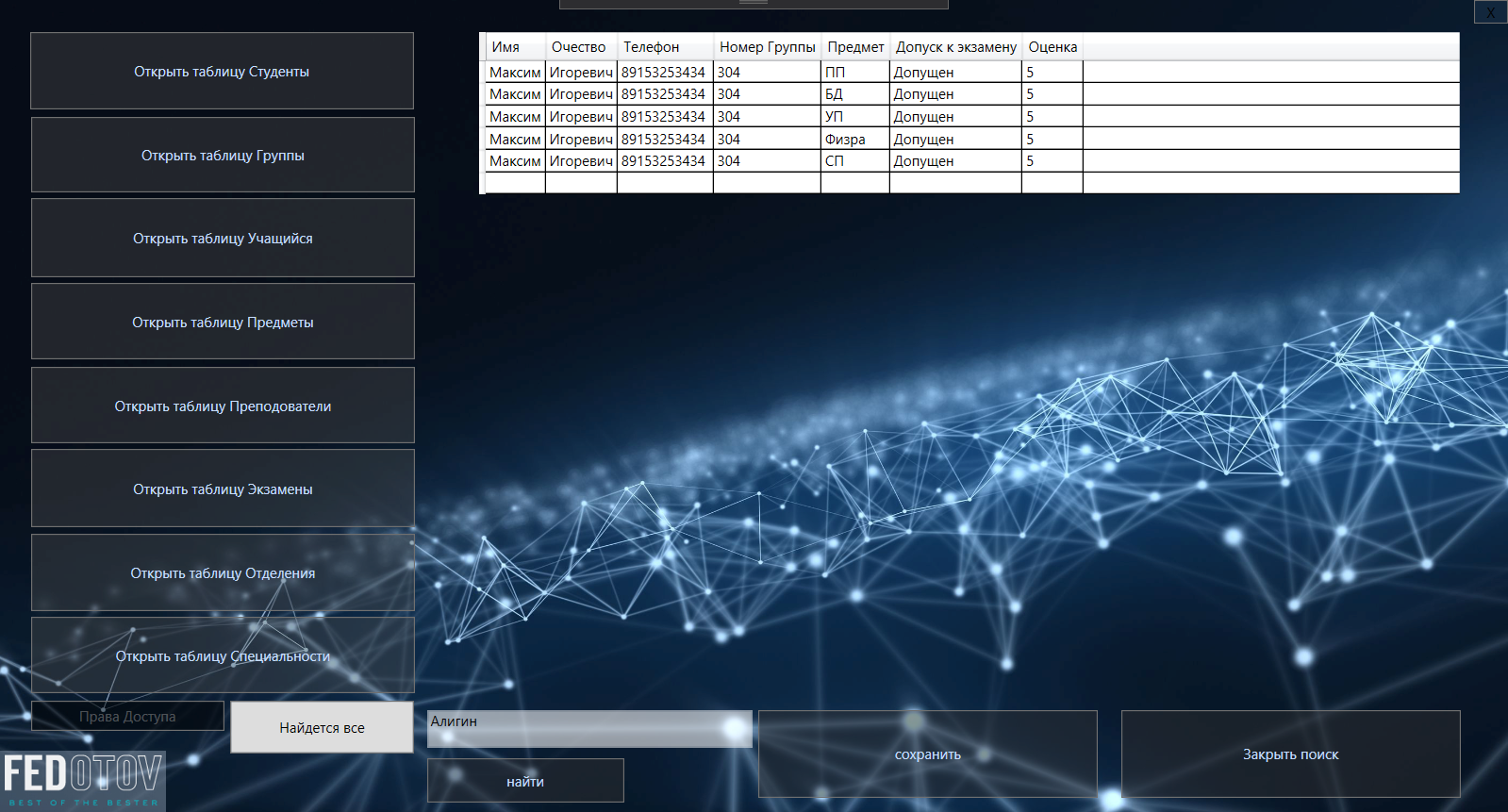


Рисунок 24 – поиск

2.6 Инструкция пользователю

Для начала использования приложения следует прочитать руководство по эксплуатации приложения и права доступа, пройти регистрацию,. После завершения процесса регистрации пользователю будут доступны все возможности, которые обеспечивает его роль. Чтобы повторно войти в приложение следует пройти авторизацию. При вводе некорректных данных будет выведено сообщение об ошибке.

2.6.1 Общие сведения об информационной системе

Информационная система разработана для базы данных колледжа. Язык интерфейса – русский.

2.6.2 Требования к техническим средствам

Необходимо иметь локальный компьютер с установленной Microsoft SQL Management Studio, в которой необходимо создать базу данных приложения используя скрипт.

2.6.3 Требования к программным средствам

Для использования программы на локальном компьютере пользователя должны быть установлены Microsoft SQL Management Studio 2017 или выше, а так-же сама программа и база данных, приложенная к ней.

2.6.4 Настройка информационной системы

Информационная система настроена для интуитивного взаимодействия с пользователем. Интерфейс будет дорабатываться и упрощаться в соответствии с критикой и в целях достижения максимального удобства.

2.6.5 Формы ввода

В приложении присутствуют такие формы ввода как:

* регистрация. Содержит поля логин, пароль,
* авторизация. Содержит поля логин и пароль,
* таблицы. Содержит стандартные DataGrid поля,
* поиск. Содержит поле “введите фамилию”.

2.6.6 Отчеты

# ЗАКЛЮЧЕНИЕ

Была проведена работа в ходе которой было спроектировано и создано приложение для редактирования информации базы данных колледжа, а также возможности добавлять информацию для осуществления этого. В ходе разработки были получены практические навыки написания информационных систем для учебных заведений. Разработанная система может быть усовершенствована с целью детальной доработки редактирования и структурирования всей доступной информации базы данных колледжа, для более точного редактирования и повышения удобства эксплуатации для конечного пользователя. Были выполнены следующие цели проекта:

* улучшение навыков проектирования баз данных и архитектуры приложений,
* создание базы данных для системы,
* написание информационной системы для колледжа.

Подводя итоги необходимо сказать, что приложение и база данных были успешно выполнены, протестированы и готовы к работе. Во время выполнения проекта возникало множество проблем, которые были успешно решены.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Руководство компании Microsoft по .NET Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/dotnet/framework/ (дата обращения 15.04.2020),
2. Руководство компании Microsoft по Entity Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/ef/ (дата обращения 05.03.2020),

# ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных, на создание учетных записей пользователей, на назначение привилегий

Все скрипты на создание и заполнение базы данных доступны по ссылке на GitHub для учебных заведений.

USE [master]

GO

/\*\*\*\*\*\* Object: Database [Students] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

CREATE DATABASE [Students]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'БД\_Студентов', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\БД\_Студентов.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'БД\_Студентов\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\БД\_Студентов\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

ALTER DATABASE [Students] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Students].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Students] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Students] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Students] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Students] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Students] SET ARITHABORT OFF

GO

ALTER DATABASE [Students] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Students] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Students] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Students] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Students] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Students] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Students] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Students] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Students] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Students] SET DISABLE\_BROKER

GO

ALTER DATABASE [Students] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Students] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Students] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Students] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Students] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Students] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Students] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Students] SET RECOVERY SIMPLE

GO

ALTER DATABASE [Students] SET MULTI\_USER

GO

ALTER DATABASE [Students] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Students] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Students] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Students] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [Students] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [Students] SET QUERY\_STORE = OFF

GO

USE [Students]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Department1] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Department1](

[NamberDepartment] [nchar](3) NOT NULL,

[Year] [date] NOT NULL,

[Manager] [nchar](10) NULL,

[idManager] [nchar](10) NULL,

[NumberGroup] [nchar](4) NULL,

CONSTRAINT [PK\_Department1] PRIMARY KEY CLUSTERED

(

[Year] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Deparyament1] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Deparyament1](

[NamberDepartment] [nchar](5) NULL,

[Year] [nchar](10) NULL,

[Manager] [nchar](45) NULL,

[NumberGroup] [nchar](5) NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[exam] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[exam](

[NamberStudent] [nchar](15) NOT NULL,

[idSubgect] [nchar](10) NOT NULL,

[Semester] [nchar](10) NULL,

[Access] [float] NULL,

[Estimation] [real] NULL,

[DateDelivery] [date] NULL,

CONSTRAINT [PK\_exam] PRIMARY KEY CLUSTERED

(

[NamberStudent] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Group] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Group](

[NumberGroup] [nchar](4) NOT NULL,

[Year] [nchar](10) NULL,

[NumberDepartments] [nchar](10) NULL,

[TypeGroup] [float] NULL,

[idCurator] [nchar](10) NULL,

[idSpecialties] [nchar](10) NULL,

[NamberStudent] [nchar](10) NULL,

CONSTRAINT [PK\_Группа] PRIMARY KEY CLUSTERED

(

[NumberGroup] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Rols] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Rols](

[idTeacher] [int] NOT NULL,

[Login] [nchar](10) NULL,

[Passwort] [nchar](10) NULL,

[NameRole] [nchar](10) NULL,

CONSTRAINT [PK\_Rols] PRIMARY KEY CLUSTERED

(

[idTeacher] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Specialty] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Specialty](

[idSpecialties] [nchar](15) NOT NULL,

[Year] [date] NOT NULL,

[NameSpec] [nchar](10) NULL,

[NumberDepartment] [nchar](1) NULL,

[NumberGroup] [nchar](4) NULL,

[Type] [float] NULL,

CONSTRAINT [PK\_Specialty] PRIMARY KEY CLUSTERED

(

[idSpecialties] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Student] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Student](

[NamberStudent] [nchar](15) NOT NULL,

[LastName] [nchar](15) NULL,

[FerstName] [nchar](15) NULL,

[MiddleName] [nchar](15) NULL,

[Bithday] [date] NULL,

[Adddress] [nchar](15) NULL,

[PhoneNumber] [nchar](12) NULL,

[PassportSeries] [nchar](4) NULL,

[PassportNamber] [nchar](6) NULL,

CONSTRAINT [PK\_Student] PRIMARY KEY CLUSTERED

(

[NamberStudent] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Students] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Students](

[Фамилия] [nchar](15) NULL,

[Имя] [nchar](15) NULL,

[Отчество] [nchar](15) NULL,

[ДР] [date] NULL,

[Адрес] [nchar](15) NULL,

[Телефон] [nchar](12) NULL,

[СерияПаспорта] [nchar](4) NULL,

[НомерПаспорта] [nchar](6) NULL

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Studying] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Studying](

[NamberStudent] [nchar](15) NOT NULL,

[NumberGroup] [nchar](4) NULL,

[Year] [date] NULL,

[Admission] [date] NULL,

[Allocation] [date] NULL,

CONSTRAINT [PK\_учащийся] PRIMARY KEY CLUSTERED

(

[NamberStudent] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Subject] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Subject](

[idSubject] [nchar](15) NOT NULL,

[nameSubject] [nchar](10) NULL,

[idSpecialties] [nchar](10) NULL,

[Time] [nchar](10) NULL,

[idTeacher] [nchar](10) NULL,

CONSTRAINT [PK\_Subject] PRIMARY KEY CLUSTERED

(

[idSubject] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[SubSpec] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[SubSpec](

[IdSubgect] [nchar](10) NOT NULL,

[idSpecialties] [nchar](15) NOT NULL,

CONSTRAINT [PK\_SubSpec] PRIMARY KEY CLUSTERED

(

[IdSubgect] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Teacher] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Teacher](

[idTeacher] [int] NOT NULL,

[LastName] [nchar](10) NULL,

[FerstName] [nchar](10) NULL,

[MiddleName] [nchar](10) NULL,

[Experience] [nchar](10) NULL,

[Discharge] [nchar](10) NULL,

[Category] [nchar](10) NULL,

[Bithday] [date] NULL,

CONSTRAINT [PK\_Teacher] PRIMARY KEY CLUSTERED

(

[idTeacher] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TS] Script Date: 04.05.2020 21:10:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TS](

[idTeacher] [nchar](15) NOT NULL,

[Year] [nchar](5) NULL,

[idSubgect] [int] NOT NULL,

[Salary] [nchar](10) NULL,

[Time] [int] NOT NULL,

CONSTRAINT [PK\_TS] PRIMARY KEY CLUSTERED

(

[idTeacher] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

INSERT [dbo].[Department1] ([NamberDepartment], [Year], [Manager], [idManager], [NumberGroup]) VALUES (N'3 ', CAST(N'2018-01-01' AS Date), N'Гусева ', N'1 ', N'304 ')

INSERT [dbo].[Department1] ([NamberDepartment], [Year], [Manager], [idManager], [NumberGroup]) VALUES (N'1 ', CAST(N'2018-01-02' AS Date), N'Миланова ', N'2 ', N'302 ')

INSERT [dbo].[Department1] ([NamberDepartment], [Year], [Manager], [idManager], [NumberGroup]) VALUES (N'2 ', CAST(N'2018-01-03' AS Date), N'Конова ', N'3 ', N'300 ')

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'1 ', N'1 ', N'6 ', 1, 5, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'10 ', N'1 ', N'6 ', 0, 0, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'11 ', N'1 ', N'6 ', 1, 3, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'12 ', N'1 ', N'6 ', 1, 3, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'13 ', N'1 ', N'6 ', 1, 3, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'2 ', N'1 ', N'6 ', 1, 5, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'3 ', N'1 ', N'6 ', 1, 4, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'4 ', N'1 ', N'6 ', 0, 0, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'5 ', N'1 ', N'6 ', 1, 3, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'6 ', N'1 ', N'6 ', 1, 4, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'7 ', N'1 ', N'6 ', 1, 5, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[exam] ([NamberStudent], [idSubgect], [Semester], [Access], [Estimation], [DateDelivery]) VALUES (N'9 ', N'1 ', N'6 ', 1, 4, CAST(N'2020-06-20' AS Date))

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'101 ', N'2019-01-01', N'1 ', 1, N'2360 ', N'360-500 ', N'6 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'102 ', N'2019-01-01', N'2 ', 1, N'2459 ', N'400-000 ', N'7 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'103 ', N'2019-01-01', N'3 ', 2, N'4356 ', N'345-900 ', N'8 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'201 ', N'2019-01-01', N'1 ', 1, N'2347 ', N'345-900 ', N'2 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'202 ', N'2019-01-01', N'2 ', 1, N'2348 ', N'345-755 ', N'3 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'203 ', N'2019-01-01', N'3 ', 1, N'2349 ', N'345-900 ', N'4 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'204 ', N'2019-01-01', N'3 ', 2, N'2350 ', N'345-755 ', N'5 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'208 ', N'2019-01-01', N'2 ', 2, N'5000 ', N'345-100 ', N'9 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'210 ', N'2019-01-01', N'1 ', 2, N'9000 ', N'300-200 ', N'10 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'304 ', N'2019-01-01', N'1 ', 1, N'2344 ', N'345-755 ', N'1 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'305 ', N'2018-01-01', N'2 ', 2, N'2345 ', N'345-900 ', N'2 ')

INSERT [dbo].[Group] ([NumberGroup], [Year], [NumberDepartments], [TypeGroup], [idCurator], [idSpecialties], [NamberStudent]) VALUES (N'306 ', N'2018-01-01', N'3 ', 1, N'2346 ', N'345-900 ', N'3 ')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'1 ', N'Алексашенков ', N'Никита ', N'Петрович ', CAST(N'2001-05-27' AS Date), N'Улица Победы ', N'89153257915 ', N'4615', N'345676')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'10 ', N'Зайев ', N'Александр ', N'Евгеньевич ', CAST(N'2001-02-02' AS Date), N'Москва ', N'81534685945 ', N'1515', N'663344')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'11 ', N'Кувшинов ', N'Михаил ', N'Валерьевич ', CAST(N'2001-04-13' AS Date), N'Москва ', N'84536579465 ', N'6615', N'774466')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'12 ', N'Кулинечев ', N'Павел ', N'Вальрьевич ', CAST(N'2001-09-30' AS Date), N'Москва ', N'89453748573 ', N'3415', N'558877')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'13 ', N'Лопаткин ', N'Виталий ', N'Владимирович ', CAST(N'2001-01-17' AS Date), N'Ногинск ', N'84653782940 ', N'8715', N'884455')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'14 ', N'Маиров ', N'Джамбулат ', N'Залимович ', CAST(N'2001-05-20' AS Date), N'Москва ', N'89153685914 ', N'8915', N'446655')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'15 ', N'Малинин ', N'Никита ', N'Сергеевич ', CAST(N'1999-04-18' AS Date), N'Москва ', N'893546710725', N'8815', N'665577')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'16 ', N'Медведева ', N'Елизовета ', N'Константиновна ', CAST(N'2001-08-25' AS Date), N'Павловский ', N'83254809856 ', N'4415', N'887799')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'17 ', N'Морозов ', N'Игорь ', N'Михайлович ', CAST(N'2001-11-30' AS Date), N'Москва ', N'89325679166 ', N'7615', N'552244')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'18 ', N'Мочалова ', N'Екатерина ', N'Вячеславовна ', CAST(N'2001-07-15' AS Date), N'Сергеев ', N'89167438061 ', N'2015', N'778866')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'19 ', N'Полозов ', N'Виталий ', N'Сергеевич ', CAST(N'2001-06-14' AS Date), N'Зеленогорск ', N'89156378692 ', N'5515', N'556644')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'2 ', N'Алигин ', N'Максим ', N'Игоревич ', CAST(N'2001-06-01' AS Date), N'Улица Дружбы 34', N'89163457693 ', N'4515', N'546787')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'20 ', N'Попов ', N'Никита ', N'Сергеевич ', CAST(N'2001-08-30' AS Date), N'Ногинск ', N'89163057824 ', N'9415', N'334466')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'21 ', N'Посысаев ', N'Владислав ', N'Вячеславович ', CAST(N'2001-10-10' AS Date), N'Томбов ', N'8918657491 ', N'9015', N'990088')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'22 ', N'Томский ', N'Николай ', N'Алексеевич ', CAST(N'2001-05-16' AS Date), N'Ногинск ', N'88453754671 ', N'4015', N'556677')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'23 ', N'Улендеев ', N'Дмитрий ', N'Юрьевич ', CAST(N'2001-08-19' AS Date), N'Неверленд ', N'893454789016', N'7716', N'887766')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'24 ', N'Устинов ', N'Сергей ', N'Сергеевич ', CAST(N'2001-09-25' AS Date), N'Ногинск ', N'89453675815 ', N'8915', N'445566')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'25 ', N'Федотов ', N'Федор ', N'Андреевич ', CAST(N'2001-09-13' AS Date), N'Подольск ', N'89153257916 ', N'4515', N'334455')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'26 ', N'Шорохов ', N'Дмитрий ', N'Валерьевич ', CAST(N'2000-02-04' AS Date), N'Ногинск ', N'89345673891 ', N'4515', N'556677')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'3 ', N'Березников ', N'Василий ', N'Павлович ', CAST(N'2001-12-05' AS Date), N'Балашиха ', N'89093638473 ', N'4315', N'657893')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'4 ', N'Большанов ', N'Сергей ', N'Алексеевич ', CAST(N'2001-05-06' AS Date), N'Москва ', N'89154783957 ', N'2315', N'456780')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'5 ', N'Борисов ', N'Иван ', N'Сергеевич ', CAST(N'2001-10-07' AS Date), N'Ногинск ', N'84937384794 ', N'1415', N'456376')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'6 ', N'Бурханов ', N'Никита ', N'Алексеевич ', CAST(N'2001-09-06' AS Date), N'Электросталь ', N'89164736547 ', N'2615', N'123456')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'7 ', N'Громов ', N'Павел ', N'Дмитриевич ', CAST(N'2001-10-12' AS Date), N'Бутово ', N'89543685936 ', N'4715', N'456789')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'8 ', N'Дулова ', N'Дарья ', N'Павловна ', CAST(N'2001-06-21' AS Date), N'Москва ', N'89132456789 ', N'2615', N'998833')

INSERT [dbo].[Student] ([NamberStudent], [LastName], [FerstName], [MiddleName], [Bithday], [Adddress], [PhoneNumber], [PassportSeries], [PassportNamber]) VALUES (N'9 ', N'Евдокимов ', N'Юрий ', N'Олегович ', CAST(N'2001-07-18' AS Date), N'Балашиха ', N'81326584967 ', N'3815', N'559900')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'100 ', N'ПП ', N'345-755 ', N'30 ', N'111 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'111 ', N'ТРЗБД ', N'345-755 ', N'50 ', N'300 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'120 ', N'УП ', N'345-755 ', N'35 ', N'222 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'123 ', N'Русский.яз', N'345-900 ', N'50 ', N'100 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'130 ', N'Математика', N'345-900 ', N'60 ', N'333 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'150 ', N'Физ-ра ', N'345-755 ', N'40 ', N'150 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'200 ', N'СП ', N'345-755 ', N'60 ', N'110 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'201 ', N'СИ ', N'300-200 ', N'30 ', N'112 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'250 ', N'Англ.яз ', N'345-755 ', N'50 ', N'160 ')

INSERT [dbo].[Subject] ([idSubject], [nameSubject], [idSpecialties], [Time], [idTeacher]) VALUES (N'350 ', N'ИБ ', N'345-900 ', N'60 ', N'170 ')

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (100, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (110, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (111, N'Гускер ', N'Александр ', N'Игоревич ', N'100 ', N'3 ', N'крутая ', CAST(N'1978-01-19' AS Date))

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (112, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (150, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (160, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (170, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (222, N'Зайцева ', N'Екатерина ', N'Васильевна', NULL, NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (300, N'Степанова ', N'Татьяна ', N'Александр ', N'10 ', NULL, NULL, NULL)

INSERT [dbo].[Teacher] ([idTeacher], [LastName], [FerstName], [MiddleName], [Experience], [Discharge], [Category], [Bithday]) VALUES (333, NULL, NULL, NULL, NULL, NULL, NULL, NULL)

ALTER TABLE [dbo].[exam] WITH CHECK ADD CONSTRAINT [FK\_exam\_Student] FOREIGN KEY([NamberStudent])

REFERENCES [dbo].[Student] ([NamberStudent])

GO

ALTER TABLE [dbo].[exam] CHECK CONSTRAINT [FK\_exam\_Student]

GO

ALTER TABLE [dbo].[Rols] WITH NOCHECK ADD CONSTRAINT [FK\_Rols\_Teacher] FOREIGN KEY([idTeacher])

REFERENCES [dbo].[Teacher] ([idTeacher])

GO

ALTER TABLE [dbo].[Rols] NOCHECK CONSTRAINT [FK\_Rols\_Teacher]

GO

ALTER TABLE [dbo].[Specialty] WITH CHECK ADD CONSTRAINT [FK\_Specialty\_Department1] FOREIGN KEY([Year])

REFERENCES [dbo].[Department1] ([Year])

GO

ALTER TABLE [dbo].[Specialty] CHECK CONSTRAINT [FK\_Specialty\_Department1]

GO

ALTER TABLE [dbo].[Specialty] WITH CHECK ADD CONSTRAINT [FK\_Specialty\_Group] FOREIGN KEY([NumberGroup])

REFERENCES [dbo].[Group] ([NumberGroup])

GO

ALTER TABLE [dbo].[Specialty] CHECK CONSTRAINT [FK\_Specialty\_Group]

GO

ALTER TABLE [dbo].[Studying] WITH CHECK ADD CONSTRAINT [FK\_Studying\_Group] FOREIGN KEY([NumberGroup])

REFERENCES [dbo].[Group] ([NumberGroup])

GO

ALTER TABLE [dbo].[Studying] CHECK CONSTRAINT [FK\_Studying\_Group]

GO

ALTER TABLE [dbo].[Studying] WITH CHECK ADD CONSTRAINT [FK\_Studying\_Student1] FOREIGN KEY([NamberStudent])

REFERENCES [dbo].[Student] ([NamberStudent])

GO

ALTER TABLE [dbo].[Studying] CHECK CONSTRAINT [FK\_Studying\_Student1]

GO

ALTER TABLE [dbo].[SubSpec] WITH CHECK ADD CONSTRAINT [FK\_SubSpec\_Specialty] FOREIGN KEY([idSpecialties])

REFERENCES [dbo].[Specialty] ([idSpecialties])

GO

ALTER TABLE [dbo].[SubSpec] CHECK CONSTRAINT [FK\_SubSpec\_Specialty]

GO

ALTER TABLE [dbo].[SubSpec] WITH CHECK ADD CONSTRAINT [FK\_SubSpec\_Subject] FOREIGN KEY([idSpecialties])

REFERENCES [dbo].[Subject] ([idSubject])

GO

ALTER TABLE [dbo].[SubSpec] CHECK CONSTRAINT [FK\_SubSpec\_Subject]

GO

ALTER TABLE [dbo].[TS] WITH CHECK ADD CONSTRAINT [FK\_TS\_exam1] FOREIGN KEY([idTeacher])

REFERENCES [dbo].[exam] ([NamberStudent])

GO

ALTER TABLE [dbo].[TS] CHECK CONSTRAINT [FK\_TS\_exam1]

GO

ALTER TABLE [dbo].[TS] WITH CHECK ADD CONSTRAINT [FK\_TS\_Subject] FOREIGN KEY([idTeacher])

REFERENCES [dbo].[Subject] ([idSubject])

GO

ALTER TABLE [dbo].[TS] CHECK CONSTRAINT [FK\_TS\_Subject]

GO

ALTER TABLE [dbo].[TS] WITH NOCHECK ADD CONSTRAINT [FK\_TS\_Teacher] FOREIGN KEY([idSubgect])

REFERENCES [dbo].[Teacher] ([idTeacher])

GO

ALTER TABLE [dbo].[TS] NOCHECK CONSTRAINT [FK\_TS\_Teacher]

GO

USE [master]

GO

ALTER DATABASE [Students] SET READ\_WRITE

GO

# ПРИЛОЖЕНИЕ Б Код программы

Весь код приложения доступен по ссылке на GitHub

<https://github.com/FedotovComToMet/FedotovCom>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Students.Pages;

using static Students.Interface;

using System.IO;

namespace Students.Pages

{

/// <summary>

/// Логика взаимодействия для MainMenu.xaml

/// </summary>

public partial class MainMenu1 : Page

{

public MainMenu1()

{

InitializeComponent();

switch (Roli)

{

case "Пользователь":

UserDataStudents.PreviewKeyDown += PreviewKeyDownHandler;

UserDataStudents.ItemsSource = SE.Student.ToList();

UserDataGrupps.ItemsSource = SE.Group.ToList();

UserDataExam.ItemsSource = SE.Exam1.ToList();

UserDataPred.ItemsSource = SE.Subject.ToList();

UserInfo.Visibility = Visibility.Visible;

AdminInfo.Visibility = Visibility.Hidden;

break;

case "Админ":

AdminDataStudents.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataGrupps.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataExam.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataUch.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataPred.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataPrepod.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataOtdel.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataCpec.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataPrava.PreviewKeyDown += PreviewKeyDownHandler;

AdminDataStudents.ItemsSource = SE.Student.ToList();

AdminDataGrupps.ItemsSource = SE.Group.ToList();

AdminDataExam.ItemsSource = SE.Exam1.ToList();

AdminDataUch.ItemsSource = SE.Studying.ToList();

AdminDataPred.ItemsSource = SE.Subject.ToList();

AdminDataPrepod.ItemsSource = SE.Teacher.ToList();

AdminDataOtdel.ItemsSource = SE.Department1.ToList();

AdminDataCpec.ItemsSource = SE.Specialty.ToList();

AdminDataPrava.ItemsSource = SE.Rols.ToList();

UserInfo.Visibility = Visibility.Hidden;

AdminInfo.Visibility = Visibility.Visible;

break;

}

}

private void open\_student(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = true;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infStud.Visibility = Visibility.Visible;

infStud1.Visibility = Visibility.Visible;

}

private void Clouse\_student(object sender, RoutedEventArgs e)

{

infStud.Visibility = Visibility.Hidden;

infStud1.Visibility = Visibility.Hidden;

}

private void open\_grup(object sender, RoutedEventArgs e){

AdminDataStudents.IsEnabled=false;

AdminDataGrupps.IsEnabled = true;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infGrupp.Visibility = Visibility.Visible;

infGrupp1.Visibility = Visibility.Visible;

}

private void clouse\_grup(object sender, RoutedEventArgs e)

{

infGrupp.Visibility = Visibility.Hidden;

infGrupp1.Visibility = Visibility.Hidden;

}

private void open\_exam(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = true;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infExam.Visibility = Visibility.Visible;

infExam1.Visibility = Visibility.Visible;

}

private void clouse\_exam(object sender, RoutedEventArgs e)

{

infExam.Visibility = Visibility.Hidden;

infExam1.Visibility = Visibility.Hidden;

}

private void open\_uch(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = true;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infUch.Visibility = Visibility.Visible;

}

private void clouse\_uch(object sender, RoutedEventArgs e)

{

infUch.Visibility = Visibility.Hidden;

}

private void open\_pred(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = true;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infPred.Visibility = Visibility.Visible;

infPred1.Visibility = Visibility.Visible;

}

private void clouse\_pred(object sender, RoutedEventArgs e)

{

infPred1.Visibility = Visibility.Hidden;

infPred.Visibility = Visibility.Hidden;

}

private void open\_prepod(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = true;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infPrepod.Visibility = Visibility.Visible;

}

private void clouse\_prepod(object sender, RoutedEventArgs e)

{

infPrepod.Visibility = Visibility.Hidden;

}

private void open\_otdel(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = true;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = false;

infOtdel.Visibility = Visibility.Visible;

}

private void clouse\_otdel(object sender, RoutedEventArgs e)

{

infOtdel.Visibility = Visibility.Hidden;

}

private void open\_cpec(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = true;

AdminDataPrava.IsEnabled = false;

infCpec.Visibility = Visibility.Visible;

}

private void clouse\_cpec(object sender, RoutedEventArgs e)

{

infCpec.Visibility = Visibility.Hidden;

}

private void open\_prava(object sender, RoutedEventArgs e)

{

AdminDataStudents.IsEnabled = false;

AdminDataGrupps.IsEnabled = false;

AdminDataExam.IsEnabled = false;

AdminDataUch.IsEnabled = false;

AdminDataPred.IsEnabled = false;

AdminDataPrepod.IsEnabled = false;

AdminDataOtdel.IsEnabled = false;

AdminDataCpec.IsEnabled = false;

AdminDataPrava.IsEnabled = true;

infPrava.Visibility = Visibility.Visible;

}

private void clouse\_prava(object sender, RoutedEventArgs e)

{

infPrava.Visibility = Visibility.Hidden;

}

private void off(object sender, RoutedEventArgs e)

{

MF.Content = new Page1();

}

private void AdminDataStudents\_DataContextChanged(object sender, DataGridCellEditEndingEventArgs e)

{

}

private void AddStudent(object sender, RoutedEventArgs e)

{

int hm = AdminDataStudents.Items.Count - SE.Student.Count();

if (hm == 2)

{

Student s = (Student)AdminDataStudents.Items[AdminDataStudents.Items.Count - 2];

s.NamberStudent = "0";

foreach (Student se in SE.Student)

{

if (Convert.ToInt32(s.NamberStudent) == Convert.ToInt32(se.NamberStudent)) s.NamberStudent = Convert.ToString(Convert.ToInt32(s.NamberStudent) + 1);

if (Convert.ToInt32(s.NamberStudent) < Convert.ToInt32(se.NamberStudent)) s.NamberStudent = Convert.ToString(Convert.ToInt32(se.NamberStudent) + 1);

}

SE.Student.Add(s);

SE.SaveChanges();

}

else

{

if (hm == 1)

{

}

else

{

}

private void SaveStudent(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void SaveGrupp(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void SaveUch(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

if (row is Studying)

{

SE.Studying.Remove((Studying)row);

}

if (row is Student s)

{

SE.Student.Remove(s);

}

if (row is Rols)

{

SE.Rols.Remove((Rols)row);

}

if (row is Teacher)

{

SE.Teacher.Remove((Teacher)row);

}

if (row is Subject)

{

SE.Subject.Remove((Subject)row);

}

if (row is Group)

{

SE.Group.Remove((Group)row);

}

if (row is Exam1)

{

SE.Exam1.Remove((Exam1)row);

}

if (row is Department1)

{

SE.Department1.Remove((Department1)row);

}

if (row is Specialty)

{

SE.Specialty.Remove((Specialty)row);

}

}

SE.SaveChanges();

}

}

private void SavePred(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler3(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Subject.Remove((Subject)row);

}

SE.SaveChanges();

}

}

private void SavePrepod(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler4(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Teacher.Remove((Teacher)row);

}

SE.SaveChanges();

}

}

private void SaveExam(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler5(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Exam1.Remove((Exam1)row);

}

SE.SaveChanges();

}

}

private void SaveOtdel(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler6(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Department1.Remove((Department1)row);

}

SE.SaveChanges();

}

}

private void SaveCpec(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler7(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Specialty.Remove((Specialty)row);

}

SE.SaveChanges();

}

}

private void SavePrava(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler8(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Rols.Remove((Rols)row);

}

SE.SaveChanges();

}

}

public class FindMarks

{

public string FerstName { get; set; }

public string MiddleName { get; set; }

public string PhoneNumber { get; set; }

public int NumberGroup { get; set; }

public string NameSubject { get; set; }

public string Access { get; set; }

public float Estimation { get; set; }

}

private void seach\_Click(object sender, RoutedEventArgs e)

{

List<FindMarks> FM = new List<FindMarks>();

string Familia = FamiliaFindUser.Text;

List<Student> FS = new List<Student>();

foreach (Student S in SE.Student)

{

if (S.LastName == Familia) FS.Add(S);

}

foreach (Student S in FS)

{

int GR = -1;

foreach (Studying ST in SE.Studying)

{

if (S.NamberStudent == ST.NamberStudent)

{

GR = (int)ST.NumberGroup; break;

}

}

foreach (Exam1 E in SE.Exam1)

{

if (S.NamberStudent == E.NamberStudent)

{

FindMarks F = new FindMarks();

foreach (Subject SU in SE.Subject)

{

if (E.idSubgect == SU.idSubject)

{

F.NameSubject = SU.nameSubject; break;

}

}

switch (E.Access)

{

case 1:

F.Access = "Допущен";

break;

case 0:

F.Access = "Опущен";

break;

}

F.Estimation = (float)E.Estimation;

F.FerstName = S.FerstName;

F.MiddleName = S.MiddleName;

F.NumberGroup = GR;

F.PhoneNumber = S.PhoneNumber;

FM.Add(F);

}

}

}

UserPoisk.ItemsSource = FM;

}

private void Save\_Exam\_User(object sender, ContextMenuEventArgs e)

{

SE.SaveChanges();

}

private void PreviewKeyDownHandler9(object sender, KeyEventArgs e)

{

var grid = (DataGrid)sender;

if (Key.Delete == e.Key)

{

foreach (var row in grid.SelectedItems)

{

SE.Exam1.Remove((Exam1)row);

}

SE.SaveChanges();

}

}

private void Clouse\_Search(object sender, RoutedEventArgs e)

{

userpoisk.Visibility = Visibility.Hidden;

}

private void Open\_poisk(object sender, RoutedEventArgs e)

{

userpoisk.Visibility = Visibility.Visible;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

{

SE.SaveChanges();

}

}

private void seach\_Click1(object sender, RoutedEventArgs e)

{

{

List<FindMarks> FM = new List<FindMarks>();

string Familia = FamiliaFindUserA.Text;

List<Student> FS = new List<Student>();

foreach (Student S in SE.Student)

{

if (S.LastName == Familia) FS.Add(S);

}

foreach (Student S in FS)

{

int GR = -1;

foreach (Studying ST in SE.Studying)

{

if (S.NamberStudent == ST.NamberStudent)

{

GR = (int)ST.NumberGroup; break;

}

}

foreach (Exam1 E in SE.Exam1)

{

if (S.NamberStudent == E.NamberStudent)

{

FindMarks F = new FindMarks();

foreach (Subject SU in SE.Subject)

{

if (E.idSubgect == SU.idSubject)

{

F.NameSubject = SU.nameSubject; break;

}

}

switch (E.Access)

{

case 1:

F.Access = "Допущен";

break;

case 0:

F.Access = "Опущен";

break;

}

F.Estimation = (float)E.Estimation;

F.FerstName = S.FerstName;

F.MiddleName = S.MiddleName;

F.NumberGroup = GR;

F.PhoneNumber = S.PhoneNumber;

FM.Add(F);

}

}

}

UserPoiskA.ItemsSource = FM;

}

}

private void Open\_poiskA(object sender, RoutedEventArgs e)

{

userpoiskAdmin.Visibility = Visibility.Visible;

}

private void Clouse\_SearchA(object sender, RoutedEventArgs e)

{

userpoiskAdmin.Visibility = Visibility.Hidden;

}

private void SavePoiskA(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

SE.SaveChanges();

}

private void print(object sender, RoutedEventArgs e)

{

StreamWriter SW = new StreamWriter("ХарактеристикаСтудента.txt");

List<FindMarks> FM;

Window W;

switch (Roli)

{

case "Пользователь":

SW.WriteLine ("Статистика по студенту: " + FamiliaFindUser.Text);

FM = (List<FindMarks>)UserPoisk.ItemsSource;

foreach(FindMarks F in FM)

{

SW.WriteLine(F.FerstName + " " + F.MiddleName + " " + F.PhoneNumber + " " + F.NumberGroup + " " + F.NameSubject + " " + F.Access + " " + F.Estimation);

}

SW.WriteLine("Дата: " + DateTime.Now);

SW.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ");

SW.Close();

MessageBox.Show("Файл создан");

W = new MailAdress();

W.Show();

break;

case "Админ":

SW.WriteLine("Статистика по студенту: " + FamiliaFindUser.Text);

FM = (List<FindMarks>)UserPoiskA.ItemsSource;

foreach (FindMarks F in FM)

{

SW.WriteLine(F.FerstName + " " + F.MiddleName + " " + F.PhoneNumber + " " + F.NumberGroup + " " + F.NameSubject + " " + F.Access + " " + F.Estimation);

}

SW.WriteLine("Дата: " + DateTime.Now);

SW.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ");

SW.Close();

MessageBox.Show("Файл создан");

W = new MailAdress();

W.Show();

break;

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Data.SqlClient;

using Students.Pages;

using static Students.Interface;

using System.IO;

namespace Students

{

public partial class Page1 : Page

{

StreamWriter SW;

StreamReader SR;

FileInfo FileInfo = new FileInfo("Auth.dll");

public Page1()

{

InitializeComponent();

if (FileInfo.Exists)

{

SR = new StreamReader("Auth.dll");

login.Text = SR.ReadLine();

Pas.Password = SR.ReadLine();

SR.Close();

}

}

private void SighIn(object sender, RoutedEventArgs e)

{

List<Rols> rols = SE.Rols.ToList();

foreach (Rols r in rols)

{

if (r.Login == login.Text && r.Password == Pas.Password)

{

MessageBox.Show("успешно");

Roli = r.NameRole;

if (readme.IsChecked == true)

{

SW = new StreamWriter("Auth.dll");

SW.WriteLine(login.Text);

SW.WriteLine(Pas.Password);

SW.Close();

}

else

{

if (FileInfo.Exists)

{

FileInfo.Delete();

}

}

MF.Navigate(new MainMenu1());

}

}

}

private void Registration(object sender, RoutedEventArgs e)

{

MF.Content = new Page2();

}

private void instruction(object sender, RoutedEventArgs e)

{

MF.Content = new Instr();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Net.Mail;

using System.Net;

namespace Students

{

/// <summary>

/// Логика взаимодействия для MailAdress.xaml

/// </summary>

public partial class MailAdress : Window

{

public MailAdress()

{

InitializeComponent();

}

private void Send(object sender, RoutedEventArgs e)

{

MailAddress from = new MailAddress("bog.ukit@mail.ru", "Дирик");

MailAddress to = new MailAddress(adresspole.Text);

MailMessage M = new MailMessage(from, to);

M.Subject = "Характеристика студента МГУТУ им. К.Г. Разумовского";

SmtpClient smtpClient = new SmtpClient("smtp.mail.ru", 587);

smtpClient.Credentials = new NetworkCredential("bog.ukit@mail.ru", "92Alisa2012");

M.Attachments.Add(new Attachment("ХарактеристикаСтудента.txt"));

smtpClient.EnableSsl = true;

smtpClient.Send(M);

this.Close();

MessageBox.Show("Успешно");

}

private void close(object sender, RoutedEventArgs e)

{

this.Close();

}

private void adresspole\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

ng System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Students.Pages;

using static Students.Interface;

namespace Students.Pages

{

/// <summary>

/// Логика взаимодействия для PravaInstr.xaml

/// </summary>

public partial class PravaInstr : Page

{

public PravaInstr()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

MF.Content = new PravaInstr();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

MF.Content = new Page1();

}

private void Back(object sender, RoutedEventArgs e)

{

MF.Content = new Instr();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Students.Pages

{

/// <summary>

/// Логика взаимодействия для Edit.xaml

/// </summary>

public partial class Edit : Page

{

public Edit()

{

InitializeComponent();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Students.Pages;

using static Students.Interface;

namespace Students.Pages

{

/// <summary>

/// Логика взаимодействия для Page2.xaml

/// </summary>

public partial class Page2 : Page

{

public Page2()

{

InitializeComponent();

}

private void Registracia(object sender, RoutedEventArgs e)

{

int id = 0;

Boolean b = false;

List<Rols> rols = SE.Rols.ToList();

foreach (Rols r in rols)

{

if (r.idTeacher > id) id = r.idTeacher;

if (r.Login == LogReg.Text && r.Password == PasReg.Password)

{

b = true;

}

}

if (b)

{

MessageBox.Show("пользователь уже зарегистрирован");

MF.Content = new MainMenu1();

}

else

{

SE.Rols.Add(new Rols

{

idTeacher = id + 1,

Login = LogReg.Text,

Password = PasReg.Password,

NameRole = "Пользователь"

}) ;

SE.SaveChanges();

Roli = "Пользователь";

MF.Content = new MainMenu1();

}

}

private void generation(object sender, RoutedEventArgs e)

{

}

private void RollBack(object sender, RoutedEventArgs e)

{

MF.Content = new Page1();

}

}

<Page x:Class="Students.Pages.MainMenu1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Students.Pages"

mc:Ignorable="d"

Title="MainMenu1">

<Grid Name="Основной" HorizontalAlignment="Left" Height="690" VerticalAlignment="Top" Width="1280" Margin="0,0,0,-0.4">

<Grid.Background>

<ImageBrush ImageSource="ФонПользователь.jpg"/>

</Grid.Background>

<Grid Name="UserInfo" HorizontalAlignment="Right" Height="714" VerticalAlignment="Top" Width="1280" Margin="0,0,0,-24" Visibility="Visible">

<Grid Name="userpoisk" Visibility="Hidden" Margin="313,0,0,0">

<DataGrid Name="UserPoisk" HorizontalAlignment="Left" Height="548" Margin="29,35,0,0" VerticalAlignment="Top" Width="898" AutoGenerateColumns="False" BorderBrush="{x:Null}">

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding FerstName}" ClipboardContentBinding="{x:Null}" Header="Имя"/>

<DataGridTextColumn Binding="{Binding MiddleName}" ClipboardContentBinding="{x:Null}" Header="Очество"/>

<DataGridTextColumn Binding="{Binding PhoneNumber}" ClipboardContentBinding="{x:Null}" Header="Телефон"/>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="Номер Группы"/>

<DataGridTextColumn Binding="{Binding NameSubject}" ClipboardContentBinding="{x:Null}" Header="Предмет"/>

<DataGridTextColumn Binding="{Binding Access}" ClipboardContentBinding="{x:Null}" Header="Допуск к экзамену"/>

<DataGridTextColumn Binding="{Binding Estimation}" ClipboardContentBinding="{x:Null}" Header="Оценка"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="сохранить" HorizontalAlignment="Left" Margin="464,601,0,0" VerticalAlignment="Top" Width="152" Height="74" Background="#7F2D2D2E" Click="Button\_Click"/>

<Button Content="Закрыть поиск" HorizontalAlignment="Left" Margin="642,601,0,0" VerticalAlignment="Top" Width="288" Height="74" Background="#7F2D2D2E" Click="Clouse\_Search"/>

<TextBox x:Name="FamiliaFindUser" HorizontalAlignment="Left" Height="32" Margin="0,601,0,0" TextWrapping="Wrap" Text="Введите фамилию" VerticalAlignment="Top" Width="294" BorderBrush="#7FFFFFFF" Background="#92FFFFFF"/>

<Button x:Name="seach" Content="найти" HorizontalAlignment="Left" Margin="61,638,0,0" VerticalAlignment="Top" Width="167" Height="37" Click="seach\_Click" Background="#7F2D2D2E"/>

<Button Content="Печать" HorizontalAlignment="Left" Margin="347,638,0,0" VerticalAlignment="Top" Width="75" Height="37" Click="print"/>

</Grid>

<Button Content="Студенты" HorizontalAlignment="Left" VerticalAlignment="Top" Width="294" Height="170" Click="open\_student" Margin="10,35,0,0" Background="#7F2D2D2E" />

<Grid HorizontalAlignment="Left" Height="661" Name="infStud1" VerticalAlignment="Top" Width="870" Margin="374,29,0,0" Visibility="Hidden">

<DataGrid Name="UserDataStudents" HorizontalAlignment="Left" Height="551" VerticalAlignment="Top" Width="870" AutoGenerateColumns="False">

<DataGrid.BorderBrush>

<ImageBrush/>

</DataGrid.BorderBrush>

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberStudent}" ClipboardContentBinding="{x:Null}" Header="№ Студента"/>

<DataGridTextColumn Binding="{Binding LastName}" ClipboardContentBinding="{x:Null}" Header="Фамилия"/>

<DataGridTextColumn Binding="{Binding FerstName}" ClipboardContentBinding="{x:Null}" Header="Имя"/>

<DataGridTextColumn Binding="{Binding MiddleName}" ClipboardContentBinding="{x:Null}" Header="Очество"/>

<DataGridTextColumn Binding="{Binding Bithday}" ClipboardContentBinding="{x:Null}" Header="ДР"/>

<DataGridTextColumn Binding="{Binding Adress}" ClipboardContentBinding="{x:Null}" Header="Адресс"/>

<DataGridTextColumn Binding="{Binding PhoneNumber}" ClipboardContentBinding="{x:Null}" Header="Телефон"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Закрыть" HorizontalAlignment="Left" VerticalAlignment="Top" Width="329" Height="52" Click="Clouse\_student" Margin="541,599,0,0" Background="#7F2D2D2E"/>

</Grid>

<Button Content="Группы" HorizontalAlignment="Left" VerticalAlignment="Top" Width="294" Height="144" Click="open\_grup" Margin="10,205,0,0" Background="#7F2D2D2E" />

<Grid HorizontalAlignment="Left" Height="651" Name="infGrupp1" VerticalAlignment="Top" Width="870" Margin="374,29,0,0" Visibility="Hidden">

<Button Content="Закрыть" HorizontalAlignment="Left" VerticalAlignment="Top" Width="329" Height="52" Click="clouse\_grup" Margin="541,599,0,0" Background="#7F2D2D2E"/>

<DataGrid Name="UserDataGrupps" HorizontalAlignment="Left" Height="593" VerticalAlignment="Top" Width="870" AutoGenerateColumns="False">

<DataGrid.BorderBrush>

<ImageBrush/>

</DataGrid.BorderBrush>

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="№ Группы" d:IsLocked="True"/>

<DataGridTextColumn Binding="{Binding Year}" ClipboardContentBinding="{x:Null}" Header="Текущий Год"/>

<DataGridTextColumn Binding="{Binding NumberDepartments}" ClipboardContentBinding="{x:Null}" Header="Отделение"/>

<DataGridTextColumn Binding="{Binding TypeGroup}" ClipboardContentBinding="{x:Null}" Header="Тип"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

<Button Content="Экзамены" HorizontalAlignment="Left" VerticalAlignment="Top" Width="294" Height="115" Click="open\_exam" Margin="10,349,0,0" Background="#7F2D2D2E"/>

<Grid HorizontalAlignment="Left" Height="661" Name="infExam1" VerticalAlignment="Top" Width="870" Margin="374,29,0,0" Visibility="Hidden" ContextMenuClosing="Save\_Exam\_User">

<Button Content="Закрыть" HorizontalAlignment="Left" VerticalAlignment="Top" Width="329" Height="52" Click="clouse\_exam" Margin="541,599,0,0" Background="#7F2D2D2E"/>

<DataGrid Name="UserDataExam" HorizontalAlignment="Left" Height="593" VerticalAlignment="Top" Width="870" AutoGenerateColumns="False">

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberStudent}" ClipboardContentBinding="{x:Null}" Header="№ Cтудента"/>

<DataGridTextColumn Binding="{Binding idExam}" ClipboardContentBinding="{x:Null}" Header="id Экзамена"/>

<DataGridTextColumn Binding="{Binding idSubgect}" ClipboardContentBinding="{x:Null}" Header="id Предмета"/>

<DataGridTextColumn Binding="{Binding Semester}" ClipboardContentBinding="{x:Null}" Header="Семестр"/>

<DataGridTextColumn Binding="{Binding Access}" ClipboardContentBinding="{x:Null}" Header="Допуск"/>

<DataGridTextColumn Binding="{Binding Estimation}" ClipboardContentBinding="{x:Null}" Header="Оценка"/>

<DataGridTextColumn Binding="{Binding DateDelivery}" ClipboardContentBinding="{x:Null}" Header="Дата Сдачи"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="305,599,0,0" VerticalAlignment="Top" Width="218" Height="52" Background="#7F2D2D2E" Click="Button\_Click\_1"/>

</Grid>

<Button Content="Предметы" HorizontalAlignment="Left" VerticalAlignment="Top" Width="294" Height="119" Click="open\_pred" Margin="10,464,0,0" Background="#7F2D2D2E"/>

<Grid HorizontalAlignment="Left" Height="661" Name="infPred1" VerticalAlignment="Top" Width="870" Margin="374,29,0,0" Visibility="Hidden">

<DataGrid Name="UserDataPred" HorizontalAlignment="Left" Height="593" VerticalAlignment="Top" Width="870" AutoGenerateColumns="False" >

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding idSubject}" ClipboardContentBinding="{x:Null}" Header="id Предмета"/>

<DataGridTextColumn Binding="{Binding nameSubject}" ClipboardContentBinding="{x:Null}" Header="Название"/>

<DataGridTextColumn Binding="{Binding idSpecialties}" ClipboardContentBinding="{x:Null}" Header="id Специальности"/>

<DataGridTextColumn Binding="{Binding Time}" ClipboardContentBinding="{x:Null}" Header="Учебные Часы"/>

<DataGridTextColumn Binding="{Binding idTeacher}" ClipboardContentBinding="{x:Null}" Header="id Учителя"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Закрыть" HorizontalAlignment="Left" VerticalAlignment="Top" Width="329" Height="52" Click="clouse\_pred" Margin="541,599,0,0" Background="#7F2D2D2E"/>

</Grid>

<Button Content="Найдется все" HorizontalAlignment="Left" Margin="138,645,0,0" VerticalAlignment="Top" Width="166" Height="45" Click="Open\_poisk"/>

</Grid>

<Grid Name="AdminInfo" HorizontalAlignment="Left" Height="690" VerticalAlignment="Top" Width="1280" Visibility="Visible">

<Grid.Background>

<ImageBrush ImageSource="1.jpg"/>

</Grid.Background>

<Button Content="Найдется все" HorizontalAlignment="Left" Margin="196,594,0,0" VerticalAlignment="Top" Width="156" Height="45" Click="Open\_poiskA"/>

<Grid Name="userpoiskAdmin" Visibility="Hidden" Margin="313,0,0,0">

<Grid.RowDefinitions>

<RowDefinition Height="26\*"/>

<RowDefinition Height="661\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<DataGrid Name="UserPoiskA" HorizontalAlignment="Left" Height="571" Margin="93,0.4,0,0" VerticalAlignment="Top" Width="834" AutoGenerateColumns="False" BorderBrush="{x:Null}" Grid.Row="1">

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding FerstName}" ClipboardContentBinding="{x:Null}" Header="Имя"/>

<DataGridTextColumn Binding="{Binding MiddleName}" ClipboardContentBinding="{x:Null}" Header="Очество"/>

<DataGridTextColumn Binding="{Binding PhoneNumber}" ClipboardContentBinding="{x:Null}" Header="Телефон"/>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="Номер Группы"/>

<DataGridTextColumn Binding="{Binding NameSubject}" ClipboardContentBinding="{x:Null}" Header="Предмет"/>

<DataGridTextColumn Binding="{Binding Access}" ClipboardContentBinding="{x:Null}" Header="Допуск к экзамену"/>

<DataGridTextColumn Binding="{Binding Estimation}" ClipboardContentBinding="{x:Null}" Header="Оценка"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="сохранить" HorizontalAlignment="Left" Margin="331,576.4,0,0" VerticalAlignment="Top" Width="288" Height="74" Background="#7F2D2D2E" Click="SavePoiskA" Grid.Row="1" Foreground="#FFC9DFFF"/>

<Button Content="Закрыть поиск" HorizontalAlignment="Left" Margin="639,576.4,0,0" VerticalAlignment="Top" Width="288" Height="74" Background="#7F2D2D2E" Click="Clouse\_SearchA" Grid.Row="1" Foreground="#FFC9DFFF"/>

<TextBox x:Name="FamiliaFindUserA" HorizontalAlignment="Left" Height="32" Margin="50,576.4,0,0" TextWrapping="Wrap" Text="Введите фамилию" VerticalAlignment="Top" Width="276" BorderBrush="#7FFFFFFF" Background="#92FFFFFF" Grid.Row="1"/>

<Button x:Name="seachA" Content="найти" HorizontalAlignment="Left" Margin="50,617.4,0,0" VerticalAlignment="Top" Width="167" Height="37" Click="seach\_Click1" Background="#7F2D2D2E" Grid.Row="1" Foreground="#FFC9DFFF" />

<Button Content="Печать" HorizontalAlignment="Left" Margin="238,617.4,0,0" Grid.Row="1" VerticalAlignment="Top" Width="88" Height="33" Click="print"/>

</Grid>

<Button Name="B" Content="Открыть таблицу Студенты" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="66" Click="open\_student" Margin="26,27,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infStud" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataStudents" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#664C4C4C" BorderBrush="#8C688CAF" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberStudent}" ClipboardContentBinding="{x:Null}" Header="№Студента" IsReadOnly="True" />

<DataGridTextColumn Binding="{Binding LastName}" ClipboardContentBinding="{x:Null}" Header="Фамилия"/>

<DataGridTextColumn Binding="{Binding FerstName}" ClipboardContentBinding="{x:Null}" Header="Имя"/>

<DataGridTextColumn Binding="{Binding MiddleName}" ClipboardContentBinding="{x:Null}" Header="Очество"/>

<DataGridTextColumn Binding="{Binding Bithday}" ClipboardContentBinding="{x:Null}" Header="ДР"/>

<DataGridTextColumn Binding="{Binding Adress}" ClipboardContentBinding="{x:Null}" Header="Адресс"/>

<DataGridTextColumn Binding="{Binding PhoneNumber}" ClipboardContentBinding="{x:Null}" Header="Телефон"/>

<DataGridTextColumn Binding="{Binding PassportSeries}" ClipboardContentBinding="{x:Null}" Header="Паспорт Серия"/>

<DataGridTextColumn Binding="{Binding PassportNumber}" ClipboardContentBinding="{x:Null}" Header="Паспорт Номер"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddStudent" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveStudent" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="Clouse\_student" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Группы" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="64" Click="open\_grup" Margin="27,99,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infGrupp" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataGrupps" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" BorderBrush="#8C4C4C4C" >

<DataGrid.Background>

<ImageBrush/>

</DataGrid.Background>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="№Группы"/>

<DataGridTextColumn Binding="{Binding Year}" ClipboardContentBinding="{x:Null}" Header="Текущий Год"/>

<DataGridTextColumn Binding="{Binding NumberDepartments}" ClipboardContentBinding="{x:Null}" Header="Отделение"/>

<DataGridTextColumn Binding="{Binding TypeGroup}" ClipboardContentBinding="{x:Null}" Header="Тип"/>

<DataGridTextColumn Binding="{Binding idCurator}" ClipboardContentBinding="{x:Null}" Header="idКуратора"/>

<DataGridTextColumn Binding="{Binding idSpecialties}" ClipboardContentBinding="{x:Null}" Header="idСпециальности"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddGrupp" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveGrupp" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_grup" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Учащийся" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="67" Click="open\_uch" Margin="27,168,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infUch" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataUch" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberStudent}" ClipboardContentBinding="{x:Null}" Header="№Студента"/>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="№Группы"/>

<DataGridTextColumn Binding="{Binding Year}" ClipboardContentBinding="{x:Null}" Header="Учебный Год"/>

<DataGridTextColumn Binding="{Binding Admission}" ClipboardContentBinding="{x:Null}" Header="Дата Зачисления"/>

<DataGridTextColumn Binding="{Binding Allocation}" ClipboardContentBinding="{x:Null}" Header="Дата Отчисления"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddUch" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveUch" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_uch" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Предметы" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="65" Click="open\_pred" Margin="27,240,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infPred" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataPred" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding idSubject}" ClipboardContentBinding="{x:Null}" Header="id Предмета"/>

<DataGridTextColumn Binding="{Binding nameSubject}" ClipboardContentBinding="{x:Null}" Header="Название"/>

<DataGridTextColumn Binding="{Binding idSpecialties}" ClipboardContentBinding="{x:Null}" Header="id Специальности"/>

<DataGridTextColumn Binding="{Binding Time}" ClipboardContentBinding="{x:Null}" Header="Учебные Часы"/>

<DataGridTextColumn Binding="{Binding idTeacher}" ClipboardContentBinding="{x:Null}" Header="id Учителя"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddPred" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SavePred" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_pred" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Преподователи" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="65" Click="open\_prepod" Margin="27,311,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infPrepod" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataPrepod" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding idTeacher}" ClipboardContentBinding="{x:Null}" Header="id Учителя"/>

<DataGridTextColumn Binding="{Binding LastName}" ClipboardContentBinding="{x:Null}" Header="Фамилия"/>

<DataGridTextColumn Binding="{Binding FerstName}" ClipboardContentBinding="{x:Null}" Header="Имя"/>

<DataGridTextColumn Binding="{Binding MiddleName}" ClipboardContentBinding="{x:Null}" Header="Очество"/>

<DataGridTextColumn Binding="{Binding Experience}" ClipboardContentBinding="{x:Null}" Header="Стаж"/>

<DataGridTextColumn Binding="{Binding Discharge}" ClipboardContentBinding="{x:Null}" Header="Разряд"/>

<DataGridTextColumn Binding="{Binding Category}" ClipboardContentBinding="{x:Null}" Header="Категория"/>

<DataGridTextColumn Binding="{Binding Bithday}" ClipboardContentBinding="{x:Null}" Header="ДР"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddPrepod" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SavePrepod" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_prepod" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Экзамены" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="66" Click="open\_exam" Margin="27,381,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infExam" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataExam" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberStudent}" ClipboardContentBinding="{x:Null}" Header="№ Cтудента"/>

<DataGridTextColumn Binding="{Binding idExam}" ClipboardContentBinding="{x:Null}" Header="id Экзамена"/>

<DataGridTextColumn Binding="{Binding idSubgect}" ClipboardContentBinding="{x:Null}" Header="id Предмета"/>

<DataGridTextColumn Binding="{Binding Semester}" ClipboardContentBinding="{x:Null}" Header="Семестр"/>

<DataGridTextColumn Binding="{Binding Access}" ClipboardContentBinding="{x:Null}" Header="Допуск"/>

<DataGridTextColumn Binding="{Binding Estimation}" ClipboardContentBinding="{x:Null}" Header="Оценка"/>

<DataGridTextColumn Binding="{Binding DateDelivery}" ClipboardContentBinding="{x:Null}" Header="Дата Сдачи"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddExam" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveExam" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_exam" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Отделения" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="65" Click="open\_otdel" Margin="27,453,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infOtdel" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataOtdel" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding NamberDepartment}" ClipboardContentBinding="{x:Null}" Header="№ Отделения"/>

<DataGridTextColumn Binding="{Binding Year}" ClipboardContentBinding="{x:Null}" Header="Учебный Год"/>

<DataGridTextColumn Binding="{Binding Manager}" ClipboardContentBinding="{x:Null}" Header="Заведующий"/>

<DataGridTextColumn Binding="{Binding idManager}" ClipboardContentBinding="{x:Null}" Header="id Заведующего"/>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="№ Группы"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddOtdel" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveOtdel" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_otdel" Background="#664C4C4C"/>

</Grid>

<Button Content="Открыть таблицу Специальности" HorizontalAlignment="Left" VerticalAlignment="Top" Width="326" Height="65" Click="open\_cpec" Margin="27,523,0,0" Background="#664C4C4C" Foreground="#FFC9DFFF"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infCpec" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataCpec" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding idSpecialties}" ClipboardContentBinding="{x:Null}" Header="id Специальности"/>

<DataGridTextColumn Binding="{Binding Year}" ClipboardContentBinding="{x:Null}" Header="Учебный Год"/>

<DataGridTextColumn Binding="{Binding NameSpec}" ClipboardContentBinding="{x:Null}" Header="Название"/>

<DataGridTextColumn Binding="{Binding NumberDepartment}" ClipboardContentBinding="{x:Null}" Header="№ Отделения"/>

<DataGridTextColumn Binding="{Binding NumberGroup}" ClipboardContentBinding="{x:Null}" Header="№ Группы"/>

<DataGridTextColumn Binding="{Binding Type}" ClipboardContentBinding="{x:Null}" Header="Тип"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddCpec" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SaveCpec" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_cpec" Background="#664C4C4C"/>

</Grid>

<Button Content="Права Доступа" HorizontalAlignment="Left" VerticalAlignment="Top" Width="164" Height="26" Click="open\_prava" Margin="27,594,0,0" Background="#82000000" Foreground="#66A0A0A0"/>

<Grid HorizontalAlignment="Left" Height="650" Name="infPrava" VerticalAlignment="Top" Width="895" Margin="374,27,0,0" Visibility="Hidden">

<DataGrid Name="AdminDataPrava" HorizontalAlignment="Left" Height="462" VerticalAlignment="Top" Width="876" AutoGenerateColumns="False" Margin="10,10,0,0" SelectionMode="Single" Background="#8C4C4C4C" >

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding idTeacher}" ClipboardContentBinding="{x:Null}" Header="id Преподавателя"/>

<DataGridTextColumn Binding="{Binding Login}" ClipboardContentBinding="{x:Null}" Header="Логин"/>

<DataGridTextColumn Binding="{Binding Password}" ClipboardContentBinding="{x:Null}" Header="Пароль"/>

<DataGridTextColumn Binding="{Binding NameRole}" ClipboardContentBinding="{x:Null}" Header="Доступ"/>

</DataGrid.Columns>

</DataGrid>

<Button Content="Добавить" HorizontalAlignment="Left" Margin="10,495,0,0" VerticalAlignment="Top" Width="210" Height="67" Click="AddPrava" Background="#664C4C4C"/>

<Button Content="Сохранить" HorizontalAlignment="Left" Margin="244,495,0,0" VerticalAlignment="Top" Width="214" Height="67" Click="SavePrava" Background="#664C4C4C"/>

<Button Content="Закрыть" HorizontalAlignment="Left" Margin="748,495,0,0" VerticalAlignment="Top" Width="137" Height="67" FontSize="25" Click="clouse\_prava" Background="#664C4C4C"/>

</Grid>

</Grid>

<Button Content="X" HorizontalAlignment="Left" Margin="1251,0,0,0" VerticalAlignment="Top" Width="29" Click="off" Background="#FF12273D"/>

</Grid>

</Page>

}

<Window x:Class="Students.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Students"

mc:Ignorable="d"

Title="MainWindow" Height="720" Width="1280" SizeToContent="WidthAndHeight" WindowStartupLocation="CenterScreen" MinHeight="720" MinWidth="1280" ResizeMode="NoResize" >

<Frame x:Name="MainFrame" NavigationUIVisibility="Hidden"/>

</Window>

<Page x:Class="Students.Page1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Students"

mc:Ignorable="d"

Title="Page1" >

<Page.Background>

<ImageBrush />

</Page.Background>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="433\*"/>

<ColumnDefinition Width="0\*"/>

<ColumnDefinition Width="443\*"/>

</Grid.ColumnDefinitions>

<Grid RenderTransformOrigin="0.497,0.287" Grid.ColumnSpan="3" Margin="0,-10,0,10">

<TextBlock HorizontalAlignment="Left" Margin="421,343,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="44" Width="166" Background="#7F2F516C" FontFamily="Trajan Pro 3" RenderTransformOrigin="0.508,0.578" FontStretch="Expanded" FontSize="35"><Run Text="Пароль"/><Run Text=":"/></TextBlock>

<TextBlock HorizontalAlignment="Left" Margin="421,281,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="52" Width="166" Background="#7F2F516C" FontFamily="Trajan Pro 3" FontSize="30"><Run FontSize="40" Text="Логин"/><Run FontSize="40" Text=":"/><LineBreak/><Run/><LineBreak/><Run/><LineBreak/><Run/><LineBreak/><Run/><LineBreak/><Run/></TextBlock>

<TextBox x:Name="login" HorizontalAlignment="Left" Height="52" Margin="643,281,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="268" CaretBrush="#FF97A0E0" FontSize="30">

<TextBox.BorderBrush>

<ImageBrush/>

</TextBox.BorderBrush>

<TextBox.Background>

<SolidColorBrush Color="#FF2F516C" Opacity="0.5"/>

</TextBox.Background>

</TextBox>

<Button Content="Войти" HorizontalAlignment="Left" Margin="421,420,0,0" VerticalAlignment="Top" Width="487" Click="SighIn" Height="74" Background="#8C2F516C" FontFamily="Trajan Pro 3" FontSize="70">

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

</Button>

<Grid Panel.ZIndex="-2" Margin="0,10,0,0" Width="1280" Height="720" >

<Grid.Background>

<ImageBrush ImageSource="Вход.jpg"/>

</Grid.Background>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="78\*"/>

<ColumnDefinition Width="77\*"/>

</Grid.ColumnDefinitions>

<Button Content="регистрация" HorizontalAlignment="Left" Margin="506,537,0,0" VerticalAlignment="Top" Width="327" Click="Registration" Height="64" Background="#841E3446" RenderTransformOrigin="0.501,0.393" FontSize="30" Grid.ColumnSpan="2">

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

</Button>

<Button Content="Ознакомьтесь с руководством по эксплотации приложения" HorizontalAlignment="Left" Margin="423,643,0,0" VerticalAlignment="Top" Width="481" Height="38" Grid.ColumnSpan="2" Background="#841E3446" Foreground="White" Click="instruction">

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

</Button>

<CheckBox Content="запомнить меня" Name="readme" HorizontalAlignment="Left" Margin="602,384,0,0" VerticalAlignment="Top" Background="#FFC00000" BorderBrush="Black" FontWeight="Bold" Grid.ColumnSpan="2"/>

</Grid>

<PasswordBox x:Name="Pas" HorizontalAlignment="Left" Margin="643,338,0,0" VerticalAlignment="Top" Width="268" Height="49" FontSize="30" Background="#7F2F516C">

<PasswordBox.BorderBrush>

<ImageBrush/>

</PasswordBox.BorderBrush>

</PasswordBox>

</Grid>

</Grid>

</Page>

<Page x:Class="Students.Pages.Page2"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Students.Pages"

mc:Ignorable="d"

d:DesignHeight="720" d:DesignWidth="1280"

Title="Page2">

<Grid Margin="-2,-2,0,2" Width="1280">

<Grid.Background>

<ImageBrush ImageSource="ФонРег.jpg"/>

</Grid.Background>

<Button x:Name="registr" Content="Зарегистрироваться" HorizontalAlignment="Left" Margin="513,343,0,0" VerticalAlignment="Top" Width="252" Height="30" Click="Registracia" FontFamily="Cooper Black" FontSize="20" Opacity="0.6" Background="#FF0071FF"/>

<TextBlock HorizontalAlignment="Left" Margin="435,227,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="36" Width="180" FontFamily="Cooper Black" TextAlignment="Center" FontSize="20" Background="#7F208BDF"><Run Text="Введите Логин"/><Run Text=":"/></TextBlock>

<TextBlock HorizontalAlignment="Left" Margin="435,283,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="32" Width="184" FontFamily="Cooper Black" TextAlignment="Center" FontSize="20" Background="#7F208BDF"><Run Text="Введите "/><Run Text="Пароль"/><Run Text=":"/></TextBlock>

<TextBlock HorizontalAlignment="Left" Margin="463,154,0,0" TextWrapping="Wrap" Text="Регистрация" VerticalAlignment="Top" Height="53" Width="333" FontFamily="Cooper Black" TextAlignment="Center" TextDecorations="{x:Null}" FontSize="40" Background="#7F208BDF"/>

<TextBox HorizontalAlignment="Left" Height="36" Margin="645,227,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="192" Name="LogReg" Background="#FF209FE0"/>

<PasswordBox HorizontalAlignment="Left" Margin="645,283,0,0" VerticalAlignment="Top" Width="192" Height="32" Name="PasReg" Background="#FF209FE0"/>

<Button Content="Вернуться Назад" HorizontalAlignment="Left" Margin="31,42,0,0" VerticalAlignment="Top" Width="138" Click="RollBack" Opacity="1" Height="33" Background="#B2494949">

<Button.BorderBrush>

<ImageBrush/>

</Button.BorderBrush>

</Button>

</Grid>

</Page>

<Window x:Class="Students.MailAdress"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Students"

mc:Ignorable="d"

Title="MailAdress" Height="250" Width="200" WindowStartupLocation="CenterScreen">

<Grid>

<Button Content="отправить" HorizontalAlignment="Left" Margin="109,140,0,0" VerticalAlignment="Top" Width="65" Click="Send"/>

<TextBox HorizontalAlignment="Left" Height="23" Margin="21,102,0,0" TextWrapping="Wrap" Text="Введите email" VerticalAlignment="Top" Width="153" TextAlignment="Center" Name="adresspole" TextChanged="adresspole\_TextChanged"/>

<Button Content="отменить" HorizontalAlignment="Left" Margin="21,140,0,0" VerticalAlignment="Top" Width="65" Click="close"/>

<TextBlock HorizontalAlignment="Left" Margin="21,32,0,0" TextWrapping="Wrap" Text="Харрактеристика Студента" VerticalAlignment="Top" Width="153" TextAlignment="Center"/>

</Grid>

</Window>