ДЕПАРТАМЕНТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ТОМСКОЙ ОБЛАСТИ

ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Специальность 09.02.07 «Информационные системы и программирование»

РАЗРАБОТКА СИСТЕМЫ УЧЁТА УСПЕВАЕМОСТИ УЧАЩИХСЯ ОБЩЕОБРАЗОВТЕЛЬНЫХ УЧРЕЖДЕНИЙ

Пояснительная записка

к курсовому проекту

КП.22.09.02.07.603.15.ПЗ

|  |  |  |
| --- | --- | --- |
| Студент  «\_\_»\_\_\_\_\_\_\_\_ 20\_\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.Г. Львов |
| Руководитель  «\_\_»\_\_\_\_\_\_\_\_ 20\_\_ г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.Ю. Маюнова |
|  |  |  |
|  |  |  |
|  |  |  |

Томск 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc106686638)

[1 ОБЩАЯ ЧАСТЬ 5](#_Toc106686639)

[1.1 Анализ предметной области 5](#_Toc106686640)

[1.2 Выбор средств и сред разработки 6](#_Toc106686641)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 12](#_Toc106686642)

[2.1 Описание требований к информационной системе 12](#_Toc106686643)

[2.2 Диаграмма вариантов использования 16](#_Toc106686644)

[2.3 Диаграмма состояний 17](#_Toc106686645)

[2.4 Схема данных 19](#_Toc106686646)

[2.5 Пользовательские сценарии 22](#_Toc106686647)

[2.6 Прототипы основных интерфейсов 23](#_Toc106686648)

[ЗАКЛЮЧЕНИЕ 29](#_Toc106686649)

[Перечень используемых источников 30](#_Toc106686650)

[Приложение A. Листинг программы 31](#_Toc106686651)

[Приложение Б. Инструкция пользователя 62](#_Toc106686652)

# Введение

Многие организации начинают переходить на цифровое хранение информации, её обработку, поиск и учет. Хранение информации в базах данных является более удобным и структурированным.

В современном мире многим отраслям требуются информационные системы, для замещения бумажной работы. Сейчас большинство отраслей переходят с заполнения и учета бумаг, на заполнение таблиц и отчетов в специализированных программах.

Электронные дневники упрощают взаимодействие между учителями и учениками и предоставляют доступ к информации об образовательном процессе в любое время.

Школьный дневник является главным документом обучающегося в образовательном процессе. Дневник выполняет задачи журнала регистрации оценок, полученных на уроках, замечаниях по поведению, а также показывает успеваемость ученика в целом.

Самым главным свойством дневника является возможность непосредственной связи между учителем и родителями ученика. Поэтому даже при сильной занятости родителя на работе, из-за которой посещение родительских собраний становится трудной задачей, учитель сможет донести необходимую информацию о ребёнке при помощи дневника.

В результате, главной целью курсового проекта, - является создание информационной системы, которая будет хранить информацию в базе данных о всех участниках образовательного процесса, предметах, оценках и расписания, тем самым упрощая взаимодействие между учителями и родителями ученика.

Задачи курсовой работы:

1. Провести анализ предметной области
2. Сформировать требования к информационной системе
3. Разработать информационную систему
4. Составить прототип пользовательского интерфейса

# ОБЩАЯ ЧАСТЬ

## Анализ предметной области

Электронный дневник – это сервис, позволяющий участникам образовательного процесса получать информацию об учебных расписаниях, домашних заданиях, текущих и итоговых оценках в режиме онлайн.

Информационная система, разработанная для общеобразовательных учреждений необходима для снижения административной нагрузки. Система должна решать основные задачи предметной области и быть удобной для пользователя. Пользователи данной системы могут получать и вносить информацию, связанную с образовательным процессом.

При создании информационной системы, пользователю должен быть предоставлен только то функционал, которым он будет пользоваться. Система должна быть удобной в работе и выдавать ожидаемы результат выполнения функций.

В системе у пользователей будут роли. В роль администратора будет входить возможность добавления информации о расписаниях, оценках и домашних заданиях. А в роль обычных пользователей будет входить возможность получения информации об образовательном процессе. У администратора есть информация о всех участниках системы и их личные данные.

Система должна обладать надёжностью, для обеспечения круглосуточной работы пользователей и обеспечивать оперативное восстановление работоспособности при сбоях.

## Выбор средств и сред разработки

Язык программирования

C# – современный объектно-ориентированный и типобезопасный язык программирования. С# имеет очень простой и понятный синтаксис, а также поддерживает компонентно-ориентированное программирование. У языка есть характерные отличительные черты: все экземпляры классов являются объектами, поддерживается инкапсуляция, полиморфизм, наследование (также множественное наследование), реализуемы такие механизмы как композицию и агрегацию. А при появлении технологии .NET Core, приложения, написанные на данном языке, являются кроссплатформенными.

C++ – высокоуровневый язык программирования общего назначения, имеющий статическую типизацию. Подходит для решения различных задач и реализации всевозможных приложений. Одним из главных преимуществ является возможность работы с памятью, благодаря этому он хорошо подходит для создания ОС и драйверов. Данный язык так же позволяет писать проекты в объектно-ориентированном стиле.

Для разработки информационной системы в данной курсовой работе был выбран язык программирования C#, т.к. данный язык создан для программ, выполняемых в среде .NET и он идеально подходит для создания настольных приложений под Windows.

UML-диаграммы

Draw Io – это бесплатное кроссплатформенное программное обеспечение для создания различных графиков, блок-схем, диаграмм и макетов. В сервисе присутствует широкий функционал и большой набор инструментов. У Draw Io очень удобный интерфейс, хорошая скорость работы и есть интеграция с Google сервисами.

Microsoft Visio – это платный векторный графический редактор, который можно использовать для создания диаграмм и блок-схем и входит в пакет приложений Microsoft Office. В данном сервисе можно использовать большое количество шаблонов, организовывать и соединять фигуры, а также изменять фигуры с текстом.

В ходе выполнения курсовой работы я выбрал Draw Io за его доступность, простоту в использовании и интерфейсе, скорость работы и наличие бесплатного доступа ко всем функциям.

Репозитории

Для хранения и изменения всех версий проекта используются репозитории. Git – это локальная система контроля версий, чаще всего используется для кода информационных систем, но так предоставляет возможность хранения различных документов. Репозиторий это хранилище файлов с историей изменений. В дальнейшем стало возможно хранить свои репозитории не локально на жестком диске, а в интернете. Существуют такие веб-сервисы для хранения репозиториев как GitHub и GitLab.

GitHub – это крупнейший веб-сервис для совместной разработки и хостинга проектов. На GitHub есть возможность контролировать разные версии кода и управлять исходным кодом. Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых. GitHub обеспечивает контроль доступа и несколько функций совместной работы: отслеживание ошибок, запросы функций, управление задачами и непрерывная интеграция.

GitLab – веб-приложение и система управления репозиториями программного кода для Git. Обладает возможностью взаимодействовать с репозиториями, управлять правами доступа и пользователями, отслеживать ошибки, автоматизировать процессы и выполнять многие другие операции. Имеется возможность подключения платного тарифа для использования дополнительных функций.

Для хранения своего проекта я использовал веб-сервис GitHub из-за большого количества функций и удобности интерфейса.

Редактор для написания кода

Для написания информационных систем на языках программирования созданы различные IDE. IDE – это среда разработки, которая используется разработчиками для создания различного программного обеспечения. Есть IDE, которые предоставляют возможность работы только с одним языком программирования, однако большинство современных IDE позволяет работать сразу с несколькими.

Visual Studio – это интегрированная среда для написания, отладки и сборки кода, а также последующей публикации приложений. Данная программа предоставляет возможность для разработки: настольных приложений (WPF), консольных приложений (Console Application), веб-приложений (ASP.NET) и ряда других служб и приложений. Visual Studio включает в себя редактор кода с поддержкой технологии IntelliSense, встроенный отладчик, веб-редактор, дизайнер классов и дизайнер схемы баз данных. Так же у Visual Studio имеется удобный менеджер пакетов NuGet.

Eclipse – это свободная интегрированная среда разработки модульных кроссплатформенных приложений. Написана на языке программирования Java и имеет цель повышения продуктивности процесса разработки программного обеспечения. Поддерживает большое количество языков программирования.

Для разработки своего проекта я использовал Visual Studio, т.к. эта программа имеет бесплатную версию и больше подходит для разработки на языке C#, чем Eclipse.

Средство разработки настольного приложения

WinForms – это устаревшая API компании Microsoft, отвечающая за создание графического интерфеса пользователя и являющаяся частью .NET Framework. Классы, реализующие API для WinForms, не зависят от языка разработки.

WPF – это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность. Для работы с WPF требуется любой .NET-совместимый язык. По сравнению со своим предшественником, Windows Forms, WPF является более производительной, т.к. используется аппаратный ускоритель графики через DirectX.

XAML – язык разметки для таких технологий как WPF и Xamarin Forms. Используется для разметки пользовательского интерфейса, привязки заданных, поддержки событий и т.д.

Для разработки клиентского приложения с графическим интерфейсом я выбрал WPF, т.к. эта платформа является более производительной и использует удобный язык разметки интерфейса XAML. Для взаимодействия приложения и базы данных я выбрал набор библиотек ADO NET.

Хранение данных

Microsoft SQL Server – система управления реляционными базами данных, разработанная компанией Microsoft. В качестве языка запросов использует Transact-SQL, который является реализацией SQL с расширениями.

PostgreSQL – это свободная объектно-реляционная система управления базами данных. Данная система базируется на языке SQL и предоставляет гибкий доступ к базам данных, их организацию и хранение.

В ходе выполнения данной курсовой работы мною была выбрана СУБД Microsoft SQL Server т.к. является бесплатной и имеет более понятный интерфейс.

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

## 2.1 Описание требований к информационной системе

2.1.1 Общее описание

Имеется информационная система “Электронный дневник”, которая позволяет пользователям просматривать и выставлять оценки, выводить и изменять список предметов, выводить и редактирова расписание за определённое число, просматривать список и добавлять пользователей, а также изменять информацию о них.

В информационной системе предусмотрена авторизация по ролям, обычный пользователь и администратор. Изначально в базе данных создается один администратор, который может добавлять новых пользователей, но если например текущий администратор уйдет из кооператива и будет удален из системы, предусмотрена функция, которая позволяет выдать роль администратора уже имеющемуся пользователю, а не только при создании нового. У администратора есть полный функционал обычного пользователя, но имеются и свои функции, не доступные обычным пользователям.

Функции пользователя:

1. Просмотр оценок

2. Просмотр расписания по дате

3. Просмотр списка предметов

4. Просмотр личной информации

Функции администратора:

1. Добавление новых пользователей в систему
2. Удаление пользователей из системы
3. Изменение роли пользователя
4. Вывод информации о всех пользователях системы
5. Редактирование персональной информации пользователей
6. Добавление новых предметов в систему
7. Выставление и изменение оценок
8. Добавление и изменение уроков в расписании

Незарегистрированным пользователям нет доступного функционала.

2.1.2 Требования к дизайну информационной системе

Информационная система разрабатывается с простым и понятным пользовательским интерфейсом. Так что порог вхождения для использования ИС очень минимальные, большинство функционала расположено в главном меню для упрощения интерфейса.

2.1.3 Требования к структуре информационной системе

Информационная система содержит следующие разделы:

1. Окно авторизации
2. Главная страница
3. Оценки
4. Расписание
5. Предметы
6. Личный кабинет
7. Список пользователей
8. Управление оценками, предметами и расписаниями
9. Добавление нового пользователя в систему
10. Панель администратора

2.1.4 Требования к хранению данных

В информационной системе присутствует ввод, хранение и обработка данных. Все данных берутся из подключенной базы данных и передаются в приложении в поля TextBox, TextBlock или DataGrid.

2.1.5 Требования к техническому обеспечению

Для работы информационной системы необходимо техническое обеспеченье со следующими характеристиками:

1. Процессор – Intel Core i5 2300. 2.8МГц
2. Оперативная память – 4 Гб (или выше)
3. Жесткий диск – 20 Гб (или выше)
4. Видеокарта – GeForce GTX 560 2Гб

2.1.6 Требования к лингвистическому обеспечению

Информационная системы полностью на русском языке.

## 2.2 Диаграмма вариантов использования

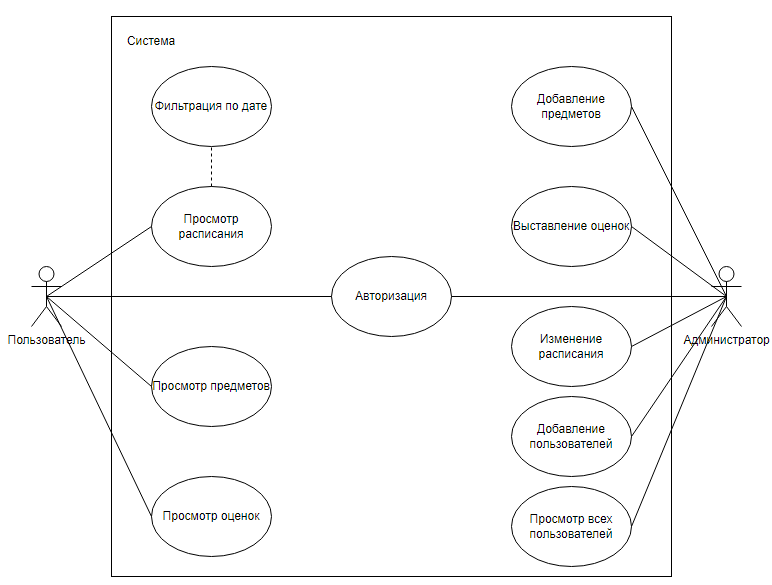
****

Рис. 1 – Диаграмма вариантов использования

## 2.3 Диаграмма состояний

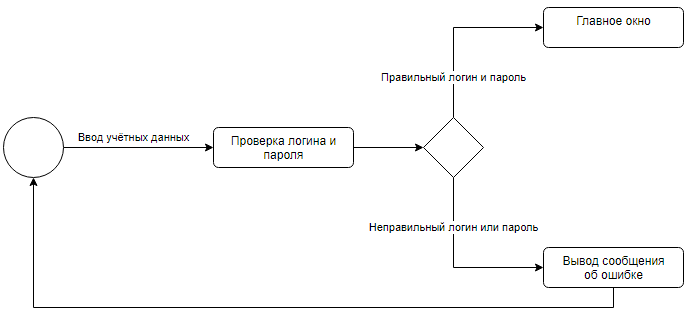
****

Рис. 2 – Диаграмма состояний авторизации

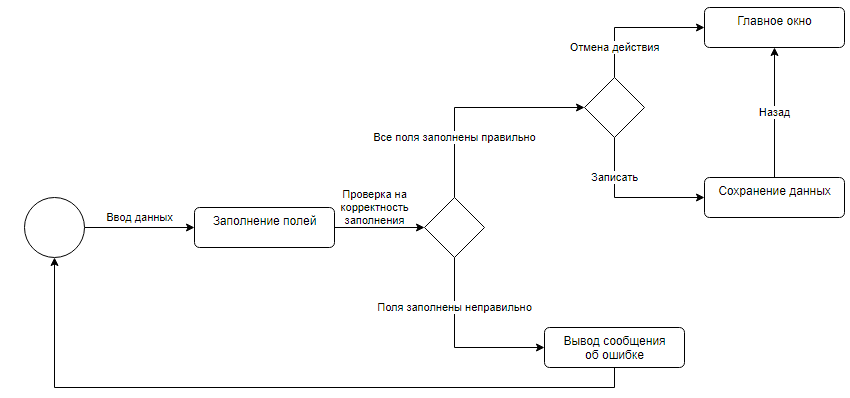
****

Рис. 3 – Диаграмма состояний добавления новых пользователей, оценок, предметов и расписаний

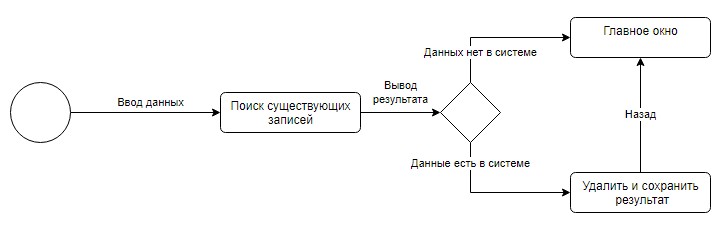
****

Рис. 4 – Диаграмма состояний поиска и удаления

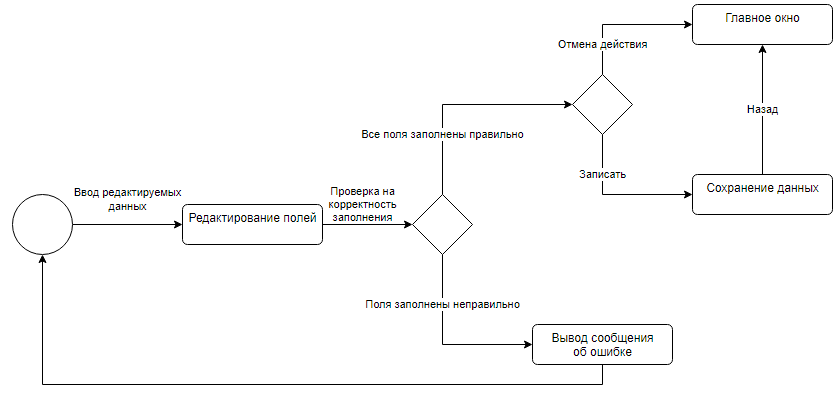
****

Рис. 5 - Диаграмма состоянийредактирования

## 2.4 Схема данных

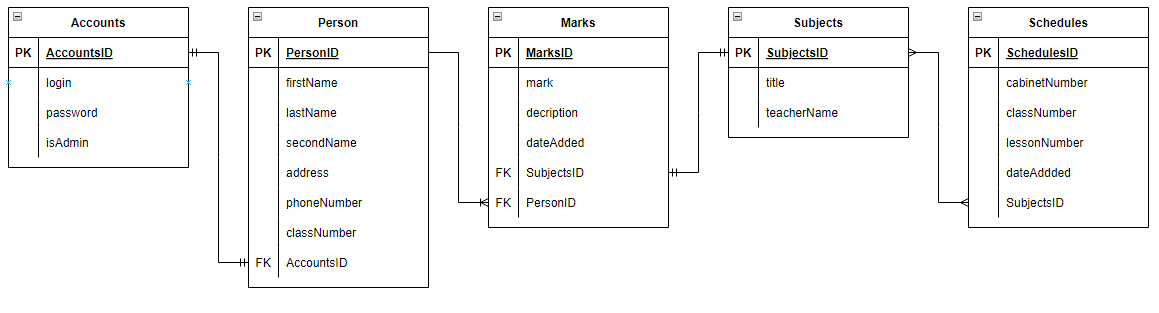
****

Рис. 6 – Схема базы данных

Таблица 1 – “Accounts”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Столбец | Тип | Комментарии |
| 1 | AccountID (PK) | Int | Идентификатор учетной записи (уникальное) |
| 2 | login | Nvarchar (30) | Логин пользователя |
| 3 | password | Nvarchar (30) | Пароль пользователя |
| 4 | isAdmin | Bit | Является ли данная учетная запись администратором |

Таблица 2 – “Person”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Столбец | Тип | Комментарии |
| 1 | PersonID (PK) | Int | Идентификатор участника (уникальное) |
| 2 | firstName | Nvarchar (50) | Имя участника |
| 3 | lastName | Nvarchar (50) | Фамилия участника |
| 4 | secondName | Nvarchar (50) | Отчество участника |
| 5 | address | Nvarchar (50) | Адрес дачного участка участника |
| 6 | phoneNumber | Nvarchar (50) | Телефонный номер |
| 7 | classNumber | Nvarchar (10) | Номер класса |
| 8 | AccountsID (FK) | Int | Номер учетной записи  (уникальное) |

Таблица 3 – “Subjects”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Столбец | Тип | Комментарии |
| 1 | SubjectsID (PK) | Int | Номер предмета (уникальное) |
| 2 | title | Nvarchar (50) | Название предмета |
| 3 | teacherName | Int | Номер участника |

Таблица 4 – “Marks”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Столбец | Тип | Комментарии |
| 1 | MarksID (PK) | Int | Номер оценки (уникальное) |
| 2 | SubjectsID (FK) | Nvarchar (50) | Название предмета |
| 3 | mark | Nvarchar (50) | Оценка |
| 4 | description | Nvarchar (200) | Описание |
| 5 | dateAdded | Date | Дата добавления |
| 6 | PersonID (FK) | int | Номер участника |

Таблица 5 – “Schedules”

|  |  |  |  |
| --- | --- | --- | --- |
| № | Столбец | Тип | Комментарии |
| 1 | SchedulesID (PK) | Int | Номер расписания (уникальное) |
| 2 | SubjectsID (FK) | Nvarchar (50) | Название предмета |
| 3 | cabinetNumber | Nvarchar (50) | Номер кабинета |
| 4 | classNumber | Nvarchar (50) | Номер класса |
| 5 | lessonNumber | Int | Номер урока |
| 6 | dateAdded | Date | Дата проведения |

## 2.5 Пользовательские сценарии

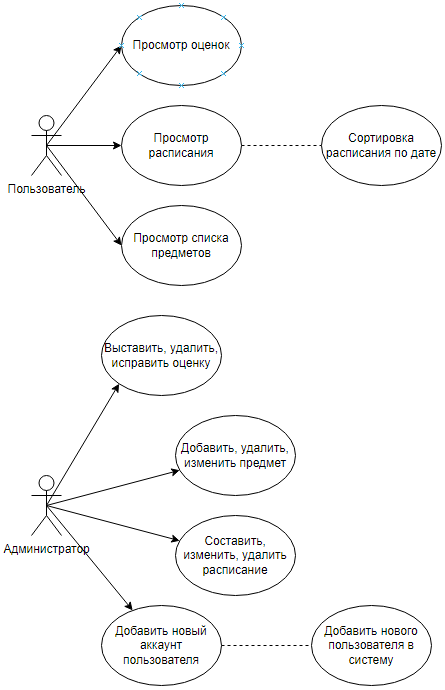


Рис. 7 – Диаграмма пользовательских сценариев

## 2.6 Прототипы основных интерфейсов

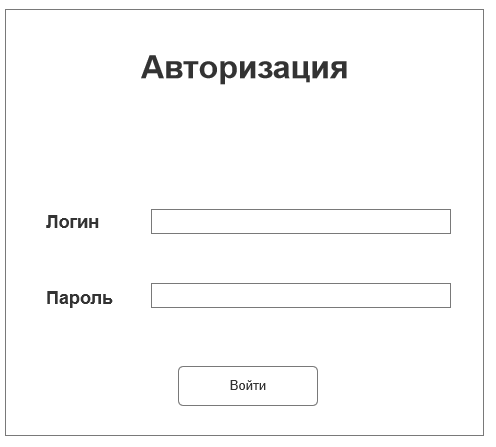
****

Рис. 8 – Окно авторизации

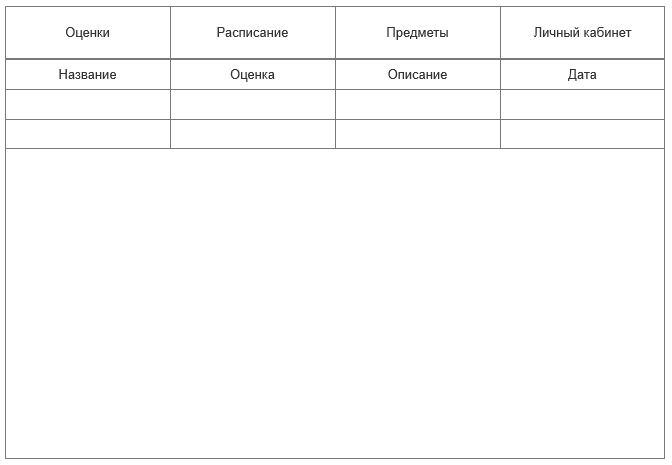


Рис. 9 – Окно оценок

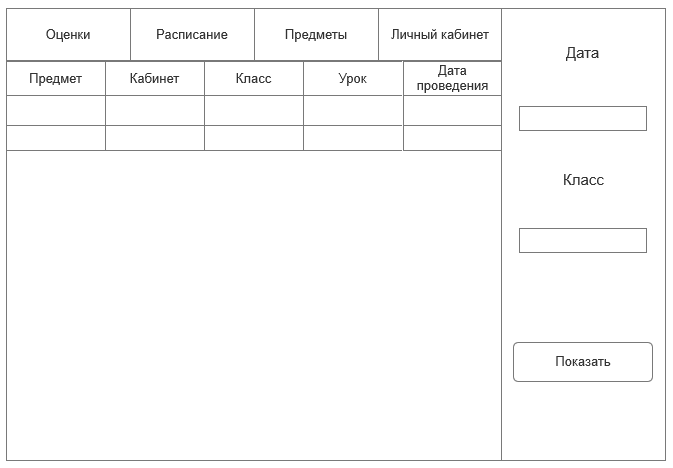
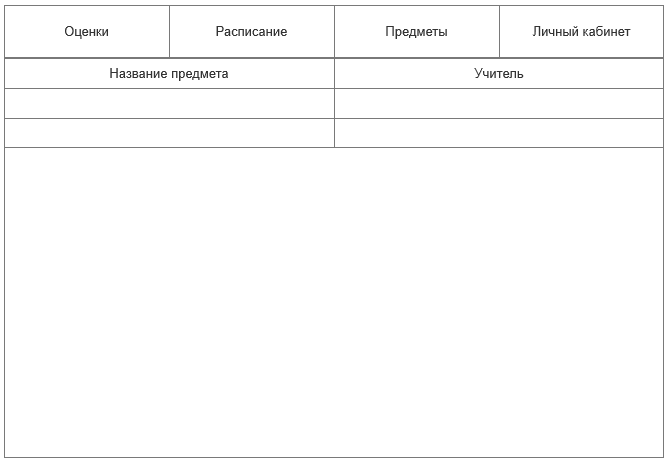
****

Рис. 10 – Окно расписания

Рис. 11 – Окно предметов

****

Рис. 12 – Окно личного кабинета

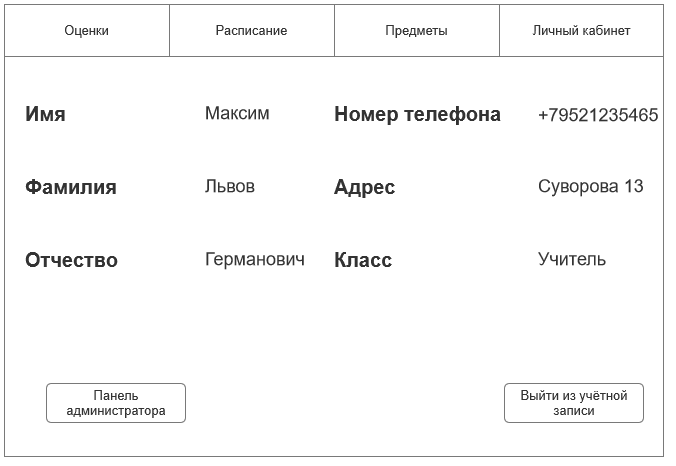
****

Рис. 13 – Окно личного кабинета с правами администратора

****

Рис. 14 – Окно панели администратора

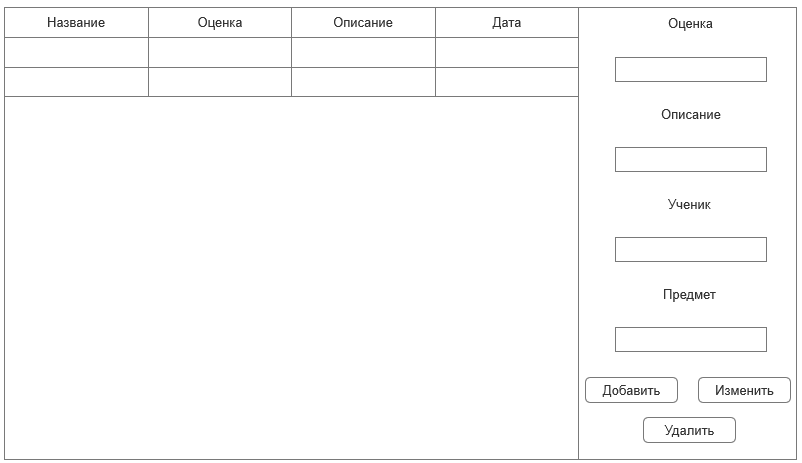
****

Рис. 15 – Окно операций с оценками

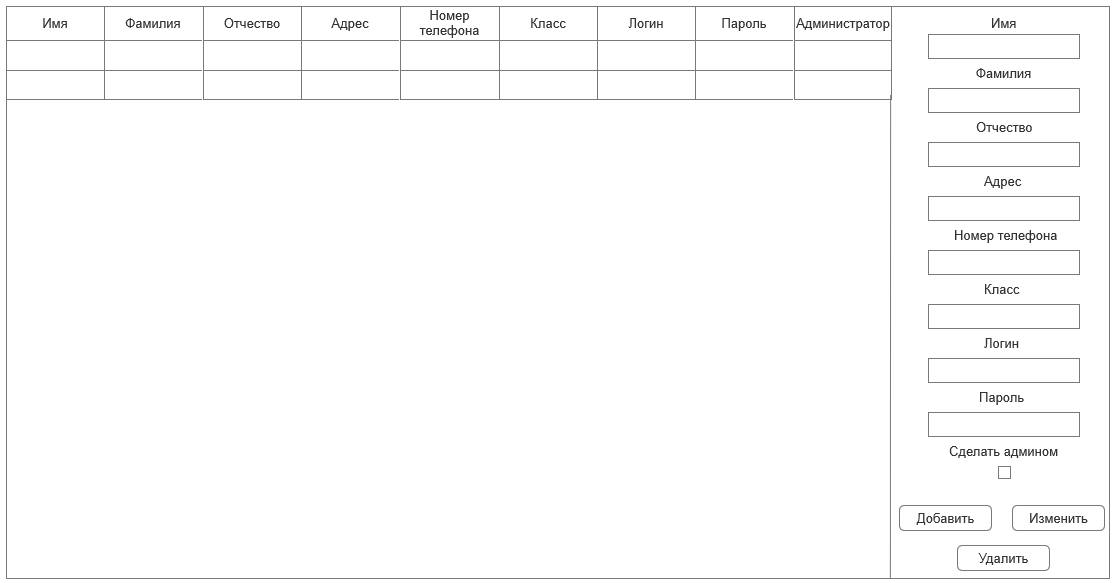
****

Рис. 16 – Окно операций с участниками

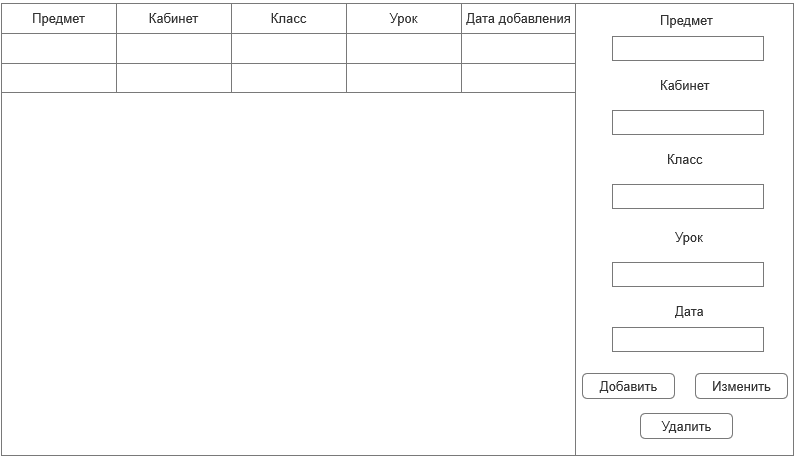
****

Рис. 17 – Окно операций с расписанием

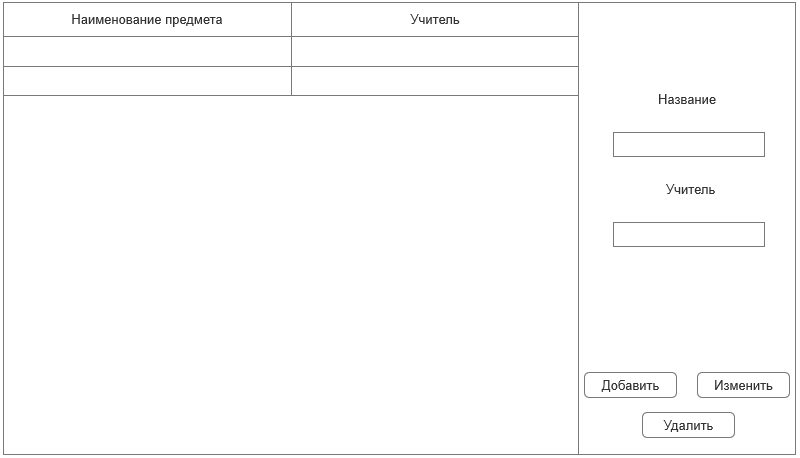
****

Рис. 18 – Окно операций с предметами

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта я проанализировал предметную область, выделил соотвествующие сущности, создал прототипы интерфейсов и разработал информационную систему, которая является настольным приложением.

Таким образом, главная цель и задачи курсового проекта по созданию информационной системы «Электронный дневник» и написании инструкции пользователя были выполнены.

# Перечень используемых источников

1. Руководство по WPF – https://metanit.com/sharp/wpf/
2. Руководство по ADO.NET и работе с базами данных – <https://metanit.com/sharp/adonetcore/>
3. Плейлист видеороликов основ C# - <https://www.youtube.com/playlist?list=PLIIXgDT0bKw4OmiZ9yGmShKsY0XncViZ8>
4. Официальная документация C# - <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

# Приложение A. Листинг программы

Класс Accounts

namespace WpfApp1

{

public class Accounts

{

public int Id { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public bool IsAdmin { get; set; }

}

}

Класс Subject

namespace WpfApp1

{

public class Subject

{

public Subject(string title, string teacherName)

{

Title = title;

TeacherName = teacherName;

}

public Subject(int id, string title, string teacherName)

{

Id = id;

Title = title;

TeacherName = teacherName;

}

public Subject(string title)

{

Title = title;

}

public int Id { get; set; }

public string Title { get; set; }

public string TeacherName { get; set; }

}

}

Класс Schedule

namespace WpfApp1

{

public class Schedule

{

public Schedule(string subjectName, string cabinetNumber, string classNumber, int lessonNumber, DateTime dateAdded)

{

SubjectName = subjectName;

CabinetNumber = cabinetNumber;

ClassNumber = classNumber;

LessonNumber = lessonNumber;

DateAdded = dateAdded;

}

public Schedule(int id, string subjectName, string cabinetNumber, string classNumber, int lessonNumber, DateTime dateAdded)

{

Id = id;

SubjectName = subjectName;

CabinetNumber = cabinetNumber;

ClassNumber = classNumber;

LessonNumber = lessonNumber;

DateAdded = dateAdded;

}

public int Id { get; set; }

public string SubjectName { get; set; }

public string CabinetNumber { get; set; }

public string ClassNumber { get; set; }

public int LessonNumber { get; set; }

public DateTime DateAdded { get; set; }

}

}

Класс Marks

namespace WpfApp1

{

public class Marks

{

public Marks(string title, string mark, string description, DateTime dateAdded, string personMark)

{

Title = title;

Mark = mark;

Description = description;

DateAdded = dateAdded;

DateAdded.ToString("dd.MM.yyyy");

PersonMark = personMark;

}

public Marks(int id, string mark, string title, string description, DateTime dateAdded, string personMark)

{

Id = id;

Title = title;

Mark = mark;

Description = description;

DateAdded = dateAdded;

DateAdded.ToString("dd.MM.yyyy");

PersonMark = personMark;

}

public int Id { get; set; }

public string Mark { get; set; }

public string Title { get; set; }

public string Description { get; set; }

public DateTime DateAdded { get; set; }

public string PersonMark { get; set; }

}

}

Класс Person

namespace WpfApp1

{

public class Person

{

public int PersonId { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string SecondName { get; set; }

public string Address { get; set; }

public string PhoneNumber { get; set; }

public string ClassNumber { get; set; }

public int AccountId { get; set; }

public Accounts Accounts = new Accounts();

public string Login { get; set; }

public string Password { get; set; }

public bool IsAdmin { get; set; }

public Person(string firstName, string lastName, string secondName, string address, string phoneNumber, string classNumber)

{

FirstName = firstName;

LastName = lastName;

SecondName = secondName;

Address = address;

PhoneNumber = phoneNumber;

ClassNumber = classNumber;

}

public Person(int id, string firstName, string lastName, string secondName, string address, string phoneNumber, string classNumber, string login, string password, bool isAdmin)

{

PersonId = id;

FirstName = firstName;

LastName = lastName;

SecondName = secondName;

Address = address;

PhoneNumber = phoneNumber;

ClassNumber = classNumber;

Login = login;

Password = password;

IsAdmin = isAdmin;

}

}

}

Класс AdminPanelWindow

namespace WpfApp1

{

public partial class AdminPanelWindow : Window

{

public AdminPanelWindow()

{

InitializeComponent();

}

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

{

MarksOperationWindow marksOperationWindow = new MarksOperationWindow();

marksOperationWindow.Show();

}

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

{

ScheduleOperationWindow scheduleOperationWindow = new ScheduleOperationWindow();

scheduleOperationWindow.Show();

}

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

{

SubjectOperationWindow subjectOperationWindow = new SubjectOperationWindow();

subjectOperationWindow.Show();

}

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

{

PersonOperationWindow personOperationWindow = new PersonOperationWindow();

personOperationWindow.Show();

}

}

}

Класс Authorization

namespace WpfApp1

{

public partial class Authorization : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

public static int currentAccountId;

bool isAdmin;

public Authorization()

{

InitializeComponent();

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

Autho(connection);

}

}

public void Autho(SqlConnection connection)

{

string sqlExpression = "SELECT \* FROM Accounts WHERE Login = '" + login.Text + "' AND Password = '" + password.Password + "'";

try

{

if (login.Text != "" && password.Password != "")

{

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

Hide();

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

MyAccountPage myAccountPage = new MyAccountPage();

MarksPage newsPage = new MarksPage();

while (reader.Read())

{

currentAccountId = (int)reader["AccountsID"];

isAdmin = (bool)reader["isAdmin"];

}

if (isAdmin)

{

myAccountPage.VisibilityAdminButton(Visibility.Visible);

}

else

{

myAccountPage.VisibilityAdminButton(Visibility.Hidden);

}

myAccountPage.accountId = currentAccountId;

mainWindow.myAccountPage = myAccountPage;

newsPage.accountId = currentAccountId;

mainWindow.marksPage = newsPage;

}

else

{

MessageBox.Show("Неправильной логин или пароль.");

}

}

else

{

MessageBox.Show("Ошибка ввода.");

}

}

catch

{

MessageBox.Show("Ошибка подключения.");

}

}

}

}

Класс SubjectOperationWindow

namespace WpfApp1

{

public partial class SubjectOperationWindow : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Subject> subjects = new List<Subject>();

public SubjectOperationWindow()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetSubjectData(connection);

}

}

public void GetSubjectData(SqlConnection connection)

{

string sqlExpression = "SELECT SubjectsID, title, CONCAT(LastName, ' ', LEFT(FirstName, 1),'. ', LEFT(SecondName, 1), '.') FROM Subjects INNER JOIN Person ON Person.PersonID = Subjects.teacherName";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

subjects.Add(new Subject(reader.GetInt32(0), reader.GetString(1), reader.GetString(2)));

}

subjectGrid.ItemsSource = subjects.ToList();

}

reader.Close();

subjects.Clear();

}

private void subjectGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (subjectGrid.SelectedItem is Subject selectedSubject)

{

TbTitle.Text = selectedSubject.Title;

TbTeacher.Text = selectedSubject.TeacherName;

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

AddRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

UpdateRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

DeleteRow(connection);

}

}

public void AddRow(SqlConnection connection)

{

string sqlExpression = $"INSERT INTO Subjects VALUES('{TbTitle.Text}', '{TbTeacher.Text}');";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetSubjectData(connection);

TbTitle.Text = "";

TbTeacher.Text = "";

}

public void UpdateRow(SqlConnection connection)

{

if (subjectGrid.SelectedItem is Subject selectedSubject)

{

string sqlExpression = $"UPDATE Subjects SET title = '{TbTitle.Text}', teacherName = '{TbTeacher.Text}' WHERE SubjectsID = '{selectedSubject.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetSubjectData(connection);

TbTitle.Text = "";

TbTeacher.Text = "";

}

}

public void DeleteRow(SqlConnection connection)

{

if (subjectGrid.SelectedItem is Subject selectedSubject)

{

string sqlExpression = $"DELETE FROM Subjects WHERE SubjectsID = '{selectedSubject.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetSubjectData(connection);

TbTitle.Text = "";

TbTeacher.Text = "";

}

}

}

}

Класс SubjectPage

namespace WpfApp1

{

public partial class SubjectPage : Page

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Subject> subjects = new List<Subject>();

public SubjectPage()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetSubjectData(connection);

}

}

public void GetSubjectData(SqlConnection connection)

{

string sqlExpression = "SELECT title, CONCAT(LastName, ' ', LEFT(FirstName, 1),'. ', LEFT(SecondName, 1), '.') FROM Subjects INNER JOIN Person ON Person.PersonID = Subjects.teacherName";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

subjects.Add(new Subject(reader.GetString(0), reader.GetString(1)));

}

subjectGrid.ItemsSource = subjects.ToList();

}

reader.Close();

subjects.Clear();

}

}

}

Класс ScheduleOperationWindow

namespace WpfApp1

{

public partial class ScheduleOperationWindow : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Schedule> schedules = new List<Schedule>();

public ScheduleOperationWindow()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetScheduleData(connection);

}

}

public void GetScheduleData(SqlConnection connection)

{

string sqlExpression = "SELECT \* FROM Schedules";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

schedules.Add(new Schedule(reader.GetInt32(0), reader.GetString(1), reader.GetString(2), reader.GetString(3), reader.GetInt32(4), reader.GetDateTime(5)));

}

scheduleGrid.ItemsSource = schedules.ToList();

}

reader.Close();

schedules.Clear();

}

private void scheduleGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (scheduleGrid.SelectedItem is Schedule selectedSchedules)

{

TbTitle.Text = selectedSchedules.SubjectName;

TbCabinet.Text = selectedSchedules.CabinetNumber;

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

AddRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

UpdateRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

DeleteRow(connection);

}

}

public void AddRow(SqlConnection connection)

{

string sqlExpression = $"INSERT INTO Schedules VALUES('{TbTitle.Text}', '{TbCabinet.Text}', '{TbClass.Text}', '{TbLesson.Text}', '{dp2.SelectedDate}');";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetScheduleData(connection);

TbTitle.Text = "";

TbCabinet.Text = "";

TbClass.Text = "";

TbLesson.Text = "";

}

public void UpdateRow(SqlConnection connection)

{

if (scheduleGrid.SelectedItem is Schedule selectedSchedule)

{

string sqlExpression = $"UPDATE Schedules SET subjectName = '{TbTitle.Text}', CabinetNumber = '{TbCabinet.Text}', classNumber = '{TbClass.Text}', lessonNumber = '{TbLesson.Text}', dateAdded = '{dp2.SelectedDate}' WHERE SchedulesID = '{selectedSchedule.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetScheduleData(connection);

TbTitle.Text = "";

TbCabinet.Text = "";

TbClass.Text = "";

TbLesson.Text = "";

}

}

public void DeleteRow(SqlConnection connection)

{

if (scheduleGrid.SelectedItem is Schedule selectedSchedule)

{

string sqlExpression = $"DELETE FROM Schedules WHERE SchedulesID = '{selectedSchedule.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetScheduleData(connection);

TbTitle.Text = "";

TbCabinet.Text = "";

TbClass.Text = "";

TbLesson.Text = "";

}

}

}

}

Класс SchedulePage

namespace WpfApp1

{

public partial class SchedulesPage : Page

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Schedule> schedules = new List<Schedule>();

public SchedulesPage()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetScheduleData(connection);

}

}

public void GetScheduleData(SqlConnection connection)

{

string sqlExpression = "SELECT SubjectName, cabinetNumber, classNumber, lessonNumber, DateAdded FROM Schedules;";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

schedules.Add(new Schedule(reader.GetString(0), reader.GetString(1), reader.GetString(2), reader.GetInt32(3), reader.GetDateTime(4)));

}

scheduleGrid.ItemsSource = schedules.ToList();

}

reader.Close();

schedules.Clear();

}

private void DatePicker\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DateTime newdate = (DateTime)((DatePicker)sender).SelectedDate;

dp1.Text = newdate.ToString();

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetScheduleDataChanged(connection);

}

}

public void GetScheduleDataChanged(SqlConnection connection)

{

string sqlExpression = $"SELECT SubjectName, cabinetNumber, classNumber, lessonNumber, DateAdded FROM Schedules WHERE DateAdded = '{dp1.Text}' AND classNumber = '{TbClass.Text}';";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

schedules.Add(new Schedule(reader.GetString(0), reader.GetString(1), reader.GetString(2), reader.GetInt32(3), reader.GetDateTime(4)));

}

scheduleGrid.ItemsSource = schedules.ToList();

}

reader.Close();

schedules.Clear();

}

}

}

Класс MainWindow

namespace WpfApp1

{

public partial class MainWindow : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

public MarksPage marksPage = new MarksPage();

public SchedulesPage schedulePage = new SchedulesPage();

public SubjectPage subjectPage = new SubjectPage();

public MyAccountPage myAccountPage = new MyAccountPage();

public MainWindow()

{

InitializeComponent();

MyFrame.Content = marksPage;

}

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

{

MyFrame.Content = marksPage;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

marksPage.GetMarksData(connection);

}

}

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

{

MyFrame.Content = schedulePage;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

schedulePage.GetScheduleData(connection);

}

}

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

{

MyFrame.Content = subjectPage;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

subjectPage.GetSubjectData(connection);

}

}

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

{

MyFrame.Content = myAccountPage;

myAccountPage.mainWindow = this;

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

myAccountPage.GetPersonData(connection);

}

}

}

}

Класс MyAccountPage

namespace WpfApp1

{

public partial class MyAccountPage : Page

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Person> person = new List<Person>();

public MainWindow mainWindow;

public int accountId = 0;

public MyAccountPage()

{

InitializeComponent();

}

public void GetPersonData(SqlConnection connection)

{

string sqlExpression = $"SELECT FirstName, LastName, SecondName, Address, PhoneNumber, ClassNumber FROM Person WHERE AccountsID = {accountId}";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

person.Add(new Person(reader.GetString(0), reader.GetString(1), reader.GetString(2), reader.GetString(3), reader.GetString(4), reader.GetString(5)));

}

TbFirstName.Text = person[0].FirstName;

TbLastName.Text = person[0].LastName;

TbSecondName.Text = person[0].SecondName;

TbPhoneNumber.Text = person[0].PhoneNumber;

TbAddress.Text = person[0].Address;

TbClassNumber.Text = person[0].ClassNumber;

}

reader.Close();

}

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

{

Authorization authorization = new Authorization();

authorization.Show();

mainWindow.Hide();

}

public void VisibilityAdminButton(Visibility visibility)

{

adminButton.Visibility = visibility;

}

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

{

AdminPanelWindow adminPanel = new AdminPanelWindow();

adminPanel.Show();

}

}

}

Класс MarksOperationWindow

namespace WpfApp1

{

public partial class MarksOperationWindow : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Marks> marks = new List<Marks>();

public MarksOperationWindow()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetMarksData(connection);

}

}

public void GetMarksData(SqlConnection connection)

{

string sqlExpression = "SELECT MarksID, Title, Mark, Description, DateAdded, CONCAT(LastName, ' ', LEFT(FirstName, 1),'. ', LEFT(SecondName, 1), '.') FROM Marks INNER JOIN Person ON Person.PersonID = Marks.personMark";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

marks.Add(new Marks(reader.GetInt32(0), reader.GetString(1), reader.GetString(2), reader.GetString(3), reader.GetDateTime(4), reader.GetString(5)));

}

marksGrid.ItemsSource = marks.ToList();

}

reader.Close();

marks.Clear();

}

private void marksGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (marksGrid.SelectedItem is Marks selectedMarks)

{

TbName.Text = selectedMarks.Title;

TbDesc.Text = selectedMarks.Description;

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

AddRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

UpdateRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

DeleteRow(connection);

}

}

public void AddRow(SqlConnection connection)

{

SqlDateTime myDateTime = DateTime.Now;

string sqlExpression = $"INSERT INTO Marks VALUES('{TbName.Text}', '{TbMark.Text}', '{TbDesc.Text}', GETDATE(), '{TbStudent.Text}');";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetMarksData(connection);

}

public void UpdateRow(SqlConnection connection)

{

if (marksGrid.SelectedItem is Marks selectedMarks)

{

string sqlExpression = $"UPDATE Marks SET Title = '{TbName.Text}', Description = '{TbDesc.Text}', Mark = '{TbMark.Text}', personMark = '{TbStudent.Text}' WHERE MarksID='{selectedMarks.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetMarksData(connection);

TbMark.Text = "";

TbName.Text = "";

TbDesc.Text = "";

TbStudent.Text = "";

}

}

public void DeleteRow(SqlConnection connection)

{

if (marksGrid.SelectedItem is Marks selectedMarks)

{

string sqlExpression = $"DELETE FROM Marks WHERE MarksID='{selectedMarks.Id}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

command.ExecuteNonQuery();

GetMarksData(connection);

TbMark.Text = "";

TbName.Text = "";

TbDesc.Text = "";

TbStudent.Text = "";

}

}

}

}

Класс MarksPage

namespace WpfApp1

{

public partial class MarksPage : Page

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

private List<Marks> marksList = new List<Marks>();

public int accountId = 0;

public MarksPage()

{

InitializeComponent();

}

public void GetMarksData(SqlConnection connection)

{

string sqlExpression = $"SELECT Title, Mark, Description, DateAdded, CONCAT(LastName, ' ', LEFT(FirstName, 1),'. ', LEFT(SecondName, 1), '.') FROM Marks INNER JOIN Person ON Person.PersonID = Marks.personMark WHERE personMark = '{accountId}'";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

marksList.Add(new Marks(reader.GetString(0), reader.GetString(1), reader.GetString(2), reader.GetDateTime(3), reader.GetString(4)));

}

marksGrid.ItemsSource = marksList.ToList();

}

reader.Close();

marksList.Clear();

}

}

}

Класс PersonOperationWindow

namespace WpfApp1

{

public partial class PersonOperationWindow : Window

{

string connectionString = "Server=DESKTOP-K7UG4KB;Database=Kursovoi;Trusted\_Connection=True;";

List<Person> personList = new List<Person>();

public PersonOperationWindow()

{

InitializeComponent();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

GetPersonData(connection);

}

}

private void personGrid\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (personGrid.SelectedItem is Person selectedPerson)

{

TbFirstName.Text = selectedPerson.FirstName;

TbLastName.Text = selectedPerson.LastName;

TbSecondName.Text = selectedPerson.SecondName;

TbAdress.Text = selectedPerson.Address;

TbPhoneNumber.Text = selectedPerson.PhoneNumber;

TbClassNumber.Text = selectedPerson.ClassNumber;

TbLogin.Text = selectedPerson.Login;

TbPassword.Text = selectedPerson.Password;

CbIsAdmin.IsChecked = selectedPerson.IsAdmin;

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

AddRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

UpdateRow(connection);

}

}

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

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

DeleteRow(connection);

}

}

public void GetPersonData(SqlConnection connection)

{

string sqlExpression = "SELECT PersonID, FirstName, LastName, SecondName, Address, PhoneNumber, ClassNumber, Login, Password, isAdmin FROM Person INNER JOIN Accounts ON Person.AccountsID = Accounts.AccountsID";

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

personList.Add(new Person(reader.GetInt32(0), reader.GetString(1), reader.GetString(2),

reader.GetString(3), reader.GetString(4), reader.GetString(5), reader.GetString(6), reader.GetString(7), reader.GetString(8), reader.GetBoolean(9)));

}

personGrid.ItemsSource = personList.ToList();

}

reader.Close();

personList.Clear();

}

public void AddRow(SqlConnection connection)

{

string sqlExpression1 = $"INSERT INTO Accounts VALUES('{TbLogin.Text}', '{TbPassword.Text}', '{CbIsAdmin.IsChecked}');";

SqlCommand command1 = new SqlCommand(sqlExpression1, connection);

command1.ExecuteNonQuery();

int accountId = default;

string sqlExpressionRead = "SELECT TOP 1 AccountsID FROM Accounts ORDER BY AccountsID DESC";

SqlCommand commandRead = new SqlCommand(sqlExpressionRead, connection);

SqlDataReader reader = commandRead.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

accountId = reader.GetInt32(0);

}

}

reader.Close();

string sqlExpression2 = $"INSERT INTO Person VALUES('{TbFirstName.Text}', '{TbLastName.Text}', '{TbSecondName.Text}', '{TbAdress.Text}', '{TbPhoneNumber.Text}', '{TbClassNumber.Text}', '{accountId}');";

SqlCommand command2 = new SqlCommand(sqlExpression2, connection);

command2.ExecuteNonQuery();

GetPersonData(connection);

TbFirstName.Text = "";

TbLastName.Text = "";

TbSecondName.Text = "";

TbAdress.Text = "";

TbPhoneNumber.Text = "";

TbClassNumber.Text = "";

TbLogin.Text = "";

TbPassword.Text = "";

CbIsAdmin.IsChecked = false;

}

public void UpdateRow(SqlConnection connection)

{

if (personGrid.SelectedItem is Person selectedPerson)

{

int accountId = default;

string sqlExpressionRead = $"SELECT Accounts.AccountsID FROM Accounts INNER JOIN Person ON Person.AccountsID = Accounts.AccountsID WHERE Person.PersonID = '{selectedPerson.PersonId}'";

SqlCommand commandRead = new SqlCommand(sqlExpressionRead, connection);

SqlDataReader reader = commandRead.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

accountId = reader.GetInt32(0);

}

}

reader.Close();

string sqlExpression1 = $"UPDATE Accounts SET Login = '{TbLogin.Text}', Password = '{TbPassword.Text}', isAdmin = '{CbIsAdmin.IsChecked}' WHERE AccountsID = '{accountId}'";

SqlCommand command1 = new SqlCommand(sqlExpression1, connection);

command1.ExecuteNonQuery();

string sqlExpression2 = $"UPDATE Person SET FirstName = '{TbFirstName.Text}', LastName = '{TbLastName.Text}', SecondName = '{TbSecondName.Text}', Address = '{TbAdress.Text}', PhoneNumber = '{TbPhoneNumber.Text}', ClassNumber = '{TbClassNumber.Text}' WHERE PersonID = '{selectedPerson.PersonId}'";

SqlCommand command2 = new SqlCommand(sqlExpression2, connection);

command2.ExecuteNonQuery();

GetPersonData(connection);

TbFirstName.Text = "";

TbLastName.Text = "";

TbSecondName.Text = "";

TbAdress.Text = "";

TbPhoneNumber.Text = "";

TbClassNumber.Text = "";

TbLogin.Text = "";

TbPassword.Text = "";

CbIsAdmin.IsChecked = false;

}

}

public void DeleteRow(SqlConnection connection)

{

if (personGrid.SelectedItem is Person selectedPerson)

{

int accountId = default;

string sqlExpressionRead = $"SELECT Accounts.AccountsID FROM Accounts INNER JOIN Person ON Person.AccountsID = Accounts.AccountsID WHERE Person.PersonID = '{selectedPerson.PersonId}'";

SqlCommand commandRead = new SqlCommand(sqlExpressionRead, connection);

SqlDataReader reader = commandRead.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

accountId = reader.GetInt32(0);

}

}

reader.Close();

string sqlExpression1 = $"DELETE FROM Person WHERE PersonID = '{selectedPerson.PersonId}'";

SqlCommand command1 = new SqlCommand(sqlExpression1, connection);

command1.ExecuteNonQuery();

string sqlExpression2 = $"DELETE FROM Accounts WHERE AccountsID = '{accountId}'";

SqlCommand command2 = new SqlCommand(sqlExpression2, connection);

command2.ExecuteNonQuery();

GetPersonData(connection);

TbFirstName.Text = "";

TbLastName.Text = "";

TbSecondName.Text = "";

TbAdress.Text = "";

TbPhoneNumber.Text = "";

TbClassNumber.Text = "";

TbLogin.Text = "";

TbPassword.Text = "";

CbIsAdmin.IsChecked = false;

}

}

}

}

XAML AdminPanelWindow

<Window x:Class="WpfApp1.AdminPanelWindow"

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:WpfApp1"

mc:Ignorable="d"

Title="AdminPanelWindow" Height="300" Width="600" ResizeMode="NoResize">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button Name="Marks" Grid.Column="0" Margin="10, 100" Content="Оценки" Click="Marks\_Click"></Button>

<Button Name="Schedules" Grid.Column="1" Margin="10, 100" Content="Расписание" Click="Schedule\_Click"></Button>

<Button Name="Subjects" Grid.Column="2" Margin="10, 100" Content="Предметы" Click="Subject\_Click"></Button>

<Button Name="Person" Grid.Column="3" Margin="10, 100" Content="Пользователи" Click="Person\_Click"></Button>

<TextBlock Grid.ColumnSpan="5" Text="Выберите объект операции" FontSize="22" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0, 20" Height="40"></TextBlock>

</Grid>

</Window>

XAML Authorization

<Window x:Class="WpfApp1.Authorization"

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:WpfApp1"

mc:Ignorable="d"

Title="Authorization" Height="450" Width="500">

<Grid>

<TextBlock HorizontalAlignment="Center" VerticalAlignment="Top" Margin="0, 60,0,0" FontSize="25">Авторизация</TextBlock>

<TextBlock HorizontalAlignment="Left" VerticalAlignment="Center" Margin="50, -80, 0,0" FontSize="25">Логин</TextBlock>

<TextBlock HorizontalAlignment="Left" VerticalAlignment="Center" Margin="50, 80, 0,0" FontSize="25">Пароль</TextBlock>

<TextBox HorizontalAlignment="Right" VerticalAlignment="Center" Margin="50, -80, 80,0" FontSize="25" Width="200" Name="login"></TextBox>

<PasswordBox HorizontalAlignment="Right" VerticalAlignment="Center" Margin="50, 80, 80,0" FontSize="25" Width="200" Name="password"></PasswordBox>

<Button HorizontalAlignment="Center" VerticalAlignment="Bottom" Height="50" Width="150" Margin="25" FontSize="25" Click="Button\_Click">Войти</Button>

</Grid>

</Window>

XAML SubjectOperationWindow

<Window x:Class="WpfApp1.SubjectOperationWindow"

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:WpfApp1"

mc:Ignorable="d"

Title="SubjectOperationWindow" Height="450" Width="900" ResizeMode="NoResize">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="400"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<DataGrid Name="subjectGrid" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="subjectGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}" Width="50"></DataGridTextColumn>

<DataGridTextColumn Header="Наименование предмета" Binding="{Binding Title}" Width="200">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Учитель" Binding="{Binding TeacherName}" Width="150"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

<Grid Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="300"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Название" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="1" Text="Учитель" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBox Grid.Column="1" VerticalAlignment="Center" Name="TbTitle" Height="26" Margin="0,0,10,0"/>

<TextBox Grid.Column="1" Grid.Row="1" VerticalAlignment="Center" Name="TbTeacher" Height="25" Margin="0,0,10,0"></TextBox>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="8" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnInsert" Margin="50, 0, 100, 0" Click="BtnInsert\_Click">Добавить</Button>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="8" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnUpdate" Margin="150, 0, 100, 0" Click="BtnUpdate\_Click">Изменить</Button>

<Button Grid.Column="1" Grid.Row="8" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnDelete" Margin="50,0,0,0" Click="BtnDelete\_Click">Удалить</Button>

</Grid>

</Grid>

</Window>

XAML SubjectPage

<Page x:Class="WpfApp1.SubjectPage"

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:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="SubjectPage">

<Grid>

<ScrollViewer>

<DataGrid Name="subjectGrid" AutoGenerateColumns="False" IsReadOnly="True" Width="800">

<DataGrid.Columns>

<DataGridTextColumn Header="Название предмета" Binding="{Binding Title}" Width="400"></DataGridTextColumn>

<DataGridTextColumn Header="Учитель" Binding="{Binding TeacherName}" Width="400"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

</Grid>

</Page>

XAML ScheduleOperationWindow

<Window x:Class="WpfApp1.ScheduleOperationWindow"

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:WpfApp1"

mc:Ignorable="d"

Title="ScheduleOperationWindow" Height="450" Width="1000" ResizeMode="NoResize">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="500"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<DataGrid Name="scheduleGrid" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="scheduleGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}"></DataGridTextColumn>

<DataGridTextColumn Header="Предмет" Binding="{Binding SubjectName}" Width="150">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Кабинет" Binding="{Binding CabinetNumber}" Width="75">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Класс" Binding="{Binding ClassNumber}" Width="75"></DataGridTextColumn>

<DataGridTextColumn Header="Урок" Binding="{Binding LessonNumber}" Width="50"></DataGridTextColumn>

<DataGridTextColumn Header="Дата добавления" Binding="{Binding DateAdded}" Width="150"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

<Grid Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="300"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Предмет" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="1" Text="Кабинет" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="2" Text="Класс" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="3" Text="Урок" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="4" Text="Дата" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBox Grid.Column="1" Grid.Row="0" VerticalAlignment="Center" Name="TbTitle" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="1" VerticalAlignment="Center" Name="TbCabinet" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="2" VerticalAlignment="Center" Name="TbClass" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="3" VerticalAlignment="Center" Name="TbLesson" Height="25"></TextBox>

<DatePicker Name ="dp2" VerticalAlignment="Top" HorizontalAlignment="Center" Width="172" Height="25" Margin="0,25,0,0" Grid.Column ="1" Grid.Row="4"/>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="5" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnInsert" Margin="50, 0, 100, 0" Click="BtnInsert\_Click">Добавить</Button>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="5" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnUpdate" Margin="150, 0, 100, 0" Click="BtnUpdate\_Click">Изменить</Button>

<Button Grid.Column="1" Grid.Row="5" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnDelete" Margin="50,0,0,0" Click="BtnDelete\_Click">Удалить</Button>

</Grid>

</Grid>

</Window>

XAML SchedulesPage

<Page x:Class="WpfApp1.SchedulesPage"

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:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="SchedulesPage">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="600"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<ScrollViewer>

<DataGrid Name="scheduleGrid" AutoGenerateColumns="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Предмет" Binding="{Binding SubjectName}" Width="150" FontWeight="Bold">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Кабинет" Binding="{Binding CabinetNumber}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Класс" Binding="{Binding ClassNumber}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Урок" Binding="{Binding LessonNumber}" Width="50"></DataGridTextColumn>

<DataGridTextColumn Header="Дата проведения" Binding="{Binding DateAdded}" Width="200"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

</Grid>

<Grid Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<TextBlock HorizontalAlignment="Center" VerticalAlignment="Top" Height="16" Margin="0,25,0,0" Width="32"><Run Language="ru-ru" Text="Дата"/></TextBlock>

<TextBlock Grid.Row="1" Text="Класс" HorizontalAlignment="Center" VerticalAlignment="Bottom" Height="16" Margin="0,25,0,0" Width="32"></TextBlock>

<DatePicker Name ="dp1" VerticalAlignment="Top" HorizontalAlignment="Center" Width="172" Height="25" Margin="0,25,0,0" Grid.Row="1"/>

<TextBox Grid.Row="2" VerticalAlignment="Center" Name="TbClass" Height="26" Margin="42,0,42,0"/>

<Button Name ="BtnShow" HorizontalAlignment="Center" VerticalAlignment="Center" Height="22" Width="80" Click="BtnShow\_Click" RenderTransformOrigin="-0.262,0.402" Grid.Row="3">Показать</Button>

</Grid>

</Grid>

</Page>

XAML MainWindow

<Window x:Class="WpfApp1.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:WpfApp1"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" ResizeMode="NoResize">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"></RowDefinition>

<RowDefinition ></RowDefinition>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Orientation="Horizontal">

<Button Name="Marks" Content="Оценки" Height="35" MinWidth="200" Click="Marks\_Click"></Button>

<Button Name="Schedules" Content="Расписание" Height="35" MinWidth="200" Click="Schedules\_Click"></Button>

<Button Name="Subjects" Content="Предметы" Height="35" MinWidth="200" Click="Subject\_Click"></Button>

<Button Name="MyAccount" Content="Личный кабинет" Height="35" MinWidth="200" Click="MyAccount\_Click"></Button>

</StackPanel>

<Frame Grid.Row="1" Name="MyFrame"></Frame>

</Grid>

</Window>

XAML MyAccountPage

<Page x:Class="WpfApp1.MyAccountPage"

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:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="MyAccountPage">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="0.30\*"></RowDefinition>

<RowDefinition Height="0.30\*"></RowDefinition>

<RowDefinition Height="0.30\*"></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Margin="10" Grid.Column="0" Grid.Row="0" FontSize="25">Имя</TextBlock>

<TextBlock Margin="10" Grid.Column="0" Grid.Row="1" FontSize="25">Фамилия</TextBlock>

<TextBlock Margin="10" Grid.Column="0" Grid.Row="2" FontSize="25">Отчество</TextBlock>

<TextBlock Margin="10" Grid.Column="1" Grid.Row="0" FontSize="25">Номер телефона</TextBlock>

<TextBlock Margin="10" Grid.Column="1" Grid.Row="1" FontSize="25">Адрес</TextBlock>

<TextBlock Margin="10" Grid.Column="1" Grid.Row="2" FontSize="25">Класс</TextBlock>

<TextBlock Name="TbFirstName" Margin="150, 10, 10, 20" Grid.Column="0" Grid.Row="0" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<TextBlock Name="TbLastName" Margin="150, 10, 10, 20" Grid.Column="0" Grid.Row="1" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<TextBlock Name="TbSecondName" Margin="150, 10, 10, 20" Grid.Column="0" Grid.Row="2" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<TextBlock Name="TbPhoneNumber" Margin="220, 10, 10, 20" Grid.Column="1" Grid.Row="0" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<TextBlock Name="TbAddress" Margin="150, 10, 10, 20" Grid.Column="1" Grid.Row="1" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<TextBlock Name="TbClassNumber" Margin="150, 10, 10, 20" Grid.Column="1" Grid.Row="2" FontSize="20" HorizontalAlignment="Left" VerticalAlignment="Center"></TextBlock>

<Button Grid.Column="1" Grid.Row="3" Height="50" Width="180" Content="Выйти из учетной записи" HorizontalAlignment="Right" VerticalAlignment="Bottom" Margin="50" Click="Button\_Click"></Button>

<Button Name="adminButton" Grid.Column="0" Grid.Row="3" Height="50" Width="180" Content="Панель администратора" HorizontalAlignment="Left" VerticalAlignment="Bottom" Margin="50" Click="adminButton\_Click"/>

</Grid>

</Page>

XAML MarksOperationWindow

<Window x:Class="WpfApp1.MarksOperationWindow"

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:WpfApp1"

mc:Ignorable="d"

Title="MarksOperationWindow" Height="450" Width="1000" ResizeMode="NoResize">

<Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<DataGrid Name="marksGrid" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="marksGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding Id}"></DataGridTextColumn>

<DataGridTextColumn Header="Оценка" Binding="{Binding Title}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Предмет" Binding="{Binding Mark}" Width="100">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Описание" Binding="{Binding Description}" Width="200">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Дата" Binding="{Binding DateAdded}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Ученик" Binding="{Binding PersonMark}" Width="100"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

<Grid Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<TextBlock Text="Оценка" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Text="Описание" HorizontalAlignment="Left" VerticalAlignment="Top" Grid.Row="1" Height="auto" Margin="100,30,0,0" Width="auto"></TextBlock>

<TextBlock Text="Ученик" HorizontalAlignment="Left" VerticalAlignment="Top" Grid.Row="2" Height="auto" Margin="100,30,0,0" Width="auto"></TextBlock>

<TextBlock Text="Предмет" HorizontalAlignment="Left" VerticalAlignment="Top" Grid.Row="3" Height="auto" Margin="100,30,0,0" Width="auto"></TextBlock>

<TextBox Grid.Column="1" VerticalAlignment="Center" Name="TbMark" Height="26" Margin="0,0,10,0"></TextBox>

<TextBox Grid.Column="1" VerticalAlignment="Top" Name="TbDesc" Grid.Row="1" Height="70" TextWrapping="Wrap" Margin="0,15,10,0"></TextBox>

<TextBox Grid.Column="1" VerticalAlignment="Top" Name="TbStudent" Grid.Row="2" Height="26" Margin="0,30,10,0"></TextBox>

<TextBox Grid.Column="1" VerticalAlignment="Top" Name="TbName" Grid.Row="3" Height="26" Margin="0,30,10,0"></TextBox>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="4" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnInsert" Margin="50, 0, 100, 0" Click="BtnInsert\_Click">Добавить</Button>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="4" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnUpdate" Margin="150, 0, 100, 0" Click="BtnUpdate\_Click">Изменить</Button>

<Button Grid.Column="1" Grid.Row="4" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnDelete" Margin="0,0,0,0" Click="BtnDelete\_Click">Удалить</Button>

</Grid>

</Grid>

</Grid>

</Window>

XAML MarksPage

<Page x:Class="WpfApp1.MarksPage"

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:WpfApp1"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="950"

Title="NewsPage">

<Grid>

<ScrollViewer>

<DataGrid Name="marksGrid" AutoGenerateColumns="False" IsReadOnly="True" Width="950">

<DataGrid.Columns>

<DataGridTextColumn Header="Название" Binding="{Binding Title}" Width="150" FontWeight="Bold">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Оценка" Binding="{Binding Mark}" Width="150" FontWeight="Bold">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Описание" Binding="{Binding Description}" Width="350">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Дата" Binding="{Binding DateAdded}" Width="150"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</ScrollViewer>

</Grid>

</Page>

XAML PersonOperationWindow

<Window x:Class="WpfApp1.PersonOperationWindow"

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:WpfApp1"

mc:Ignorable="d"

Title="PersonOperationWindow" Height="500" Width="1600" ResizeMode="NoResize">

<Grid>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1100"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<DataGrid Name="personGrid" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="personGrid\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Id" Binding="{Binding PersonId}"></DataGridTextColumn>

<DataGridTextColumn Header="Имя" Binding="{Binding FirstName}" Width="100">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Фамилия" Binding="{Binding LastName}" Width="150">

<DataGridTextColumn.ElementStyle>

<Style>

<Setter Property="TextBlock.TextWrapping" Value="Wrap" />

</Style>

</DataGridTextColumn.ElementStyle>

</DataGridTextColumn>

<DataGridTextColumn Header="Отчество" Binding="{Binding SecondName}" Width="150"></DataGridTextColumn>

<DataGridTextColumn Header="Адрес" Binding="{Binding Address}" Width="150"></DataGridTextColumn>

<DataGridTextColumn Header="Номер телефона" Binding="{Binding PhoneNumber}" Width="150"></DataGridTextColumn>

<DataGridTextColumn Header="Класс" Binding="{Binding ClassNumber}" Width="75"></DataGridTextColumn>

<DataGridTextColumn Header="Логин" Binding="{Binding Login}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Пароль" Binding="{Binding Password}" Width="100"></DataGridTextColumn>

<DataGridTextColumn Header="Администратор" Binding="{Binding IsAdmin}" Width="100"></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

<Grid Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="300"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

<RowDefinition Height="20\*"/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Имя" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="1" Text="Фамилия" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="2" Text="Отчество" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="3" Text="Адрес" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock TextWrapping="Wrap" Grid.Row="4" Text="Номер телефона" HorizontalAlignment="Left" VerticalAlignment="Center" Height="36" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="5" Text="Класс" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="6" Text="Логин" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock Grid.Row="7" Text="Пароль" HorizontalAlignment="Left" VerticalAlignment="Center" Height="16" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBlock TextWrapping="Wrap" Grid.Row="8" Text="Сделать админом" HorizontalAlignment="Left" VerticalAlignment="Center" Height="36" Margin="100,0,0,0" Width="83"></TextBlock>

<TextBox Grid.Column="1" Grid.Row="0" VerticalAlignment="Center" Name="TbFirstName" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="1" VerticalAlignment="Center" Name="TbLastName" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="2" VerticalAlignment="Center" Name="TbSecondName" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="3" VerticalAlignment="Center" Name="TbAdress" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="4" VerticalAlignment="Center" Name="TbPhoneNumber" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="5" VerticalAlignment="Center" Name="TbClassNumber" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="6" VerticalAlignment="Center" Name="TbLogin" Height="25"></TextBox>

<TextBox Grid.Column="1" Grid.Row="7" VerticalAlignment="Center" Name="TbPassword" Height="25"></TextBox>

<CheckBox Grid.Column="1" Grid.Row="8" VerticalAlignment="Center" HorizontalAlignment="Left" Name="CbIsAdmin" Height="auto"></CheckBox>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="9" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnInsert" Margin="50, 0, 100, 0" Click="BtnInsert\_Click">Добавить</Button>

<Button Grid.ColumnSpan="2" Grid.Column="0" Grid.Row="9" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnUpdate" Margin="150, 0, 100, 0" Click="BtnUpdate\_Click">Изменить</Button>

<Button Grid.Column="1" Grid.Row="9" Height="35" Width="70" HorizontalAlignment="Left" VerticalAlignment="Center" Name="BtnDelete" Margin="50,0,0,0" Click="BtnDelete\_Click">Удалить</Button>

</Grid>

</Grid>

</Grid>

</Window>

# Приложение Б. Инструкция пользователя

1) Для запуска приложения нужно запустить файл WpfApp1.exe

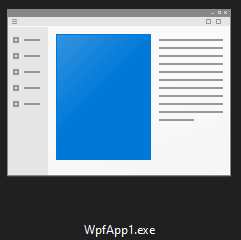


Рис. 19 – Ярлык приложения

2) При запуске появится окно, в котором нужно ввести свой логин, пароль и нажать кнопку “Войти”.

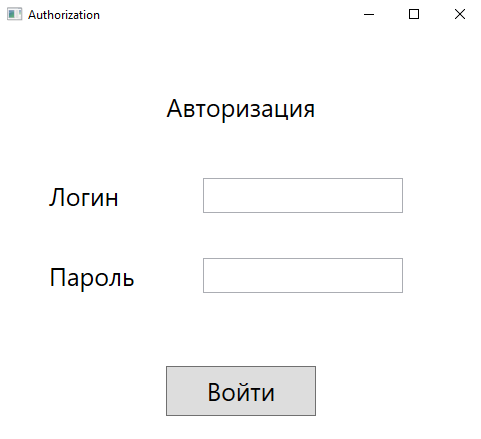


Рис. 20 – Окно авторизации

3) После авторизации откроется окно приложение и первая вкладка “Оценки”.

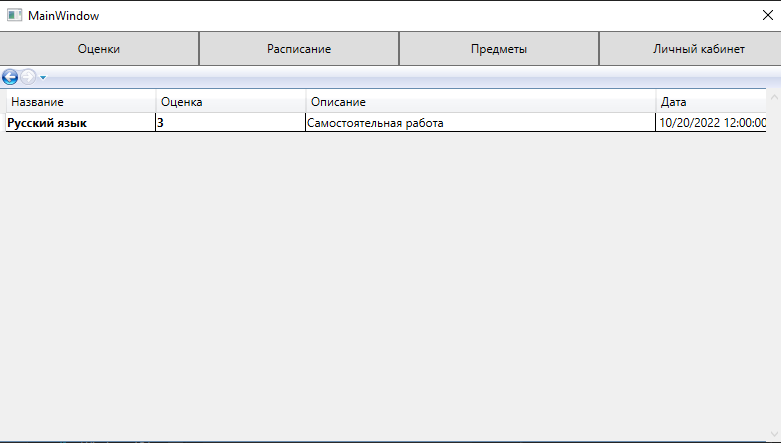


Рис. 21 – Окно оценок

4) Нажав на кнопку “Расписание” пользователь попадает в соответствующее окно и может изменять дату и класс для фильтрации расписания.

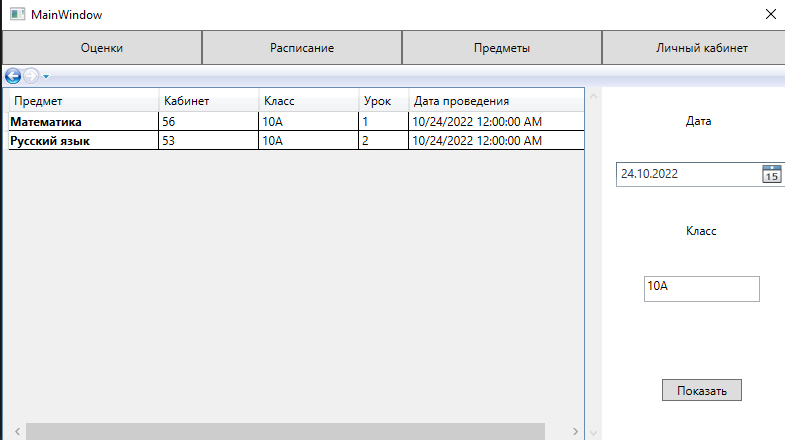


Рис. 22 – Окно расписания

5) Нажав на кнопку “Предметы” пользователю открывается окно с имеющимися на данный момент предметами.

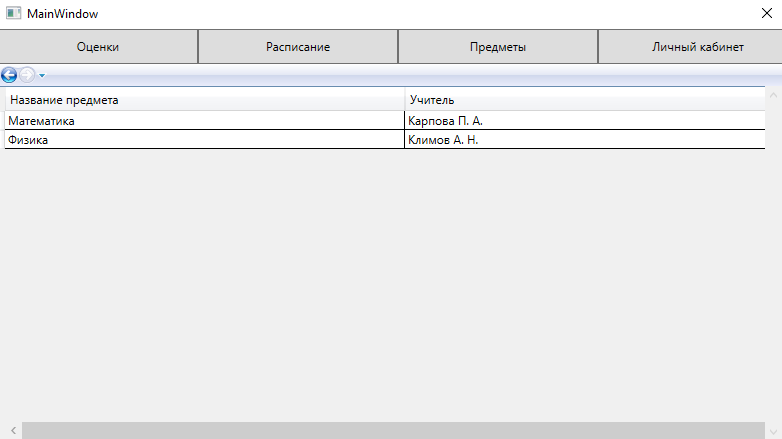


Рис. 23 – Окно предметов

6) Нажав на кнопку “Личный кабинет” пользователю открывается окно, в котором указана вся его информация и имеется кнопка “Выйти из учетной записи” для выхода из системы.

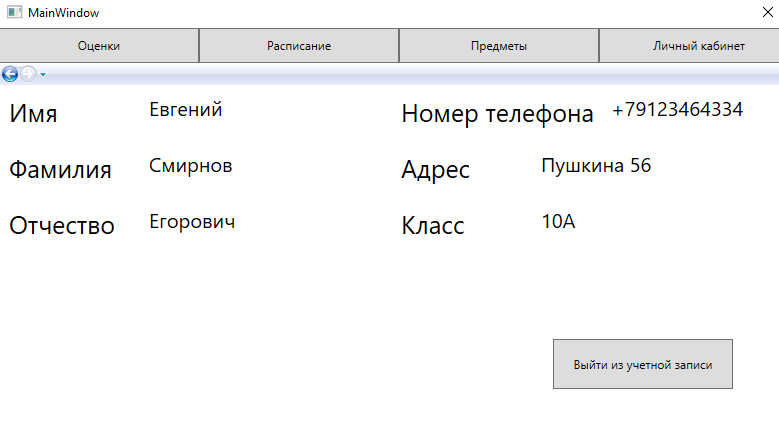


Рис. 24 – Окно личного кабинета пользователя

7) Если у пользователя есть права администратора, то в окне личного кабинета появится новая кнопка “Панель администратора”.

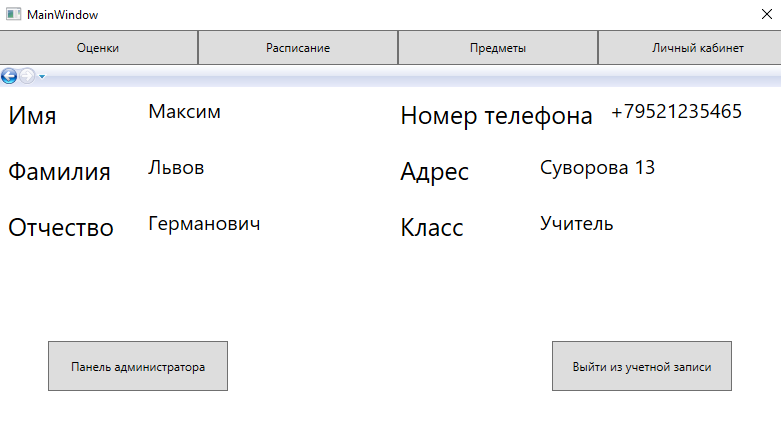


Рис. 25 – Окно личного кабинета администратора

8) В панели администратора представлены кнопки, открывающие соотвествующие разделы редактирования.

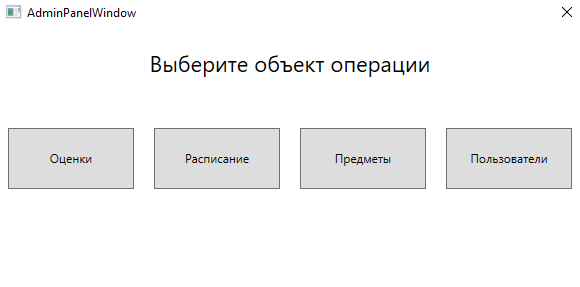


Рис. 26 – Окно выбора операций

9) Для добавления оценки необходимо заполнить соответствующие поля и нажать на кнопку “Добавить”.

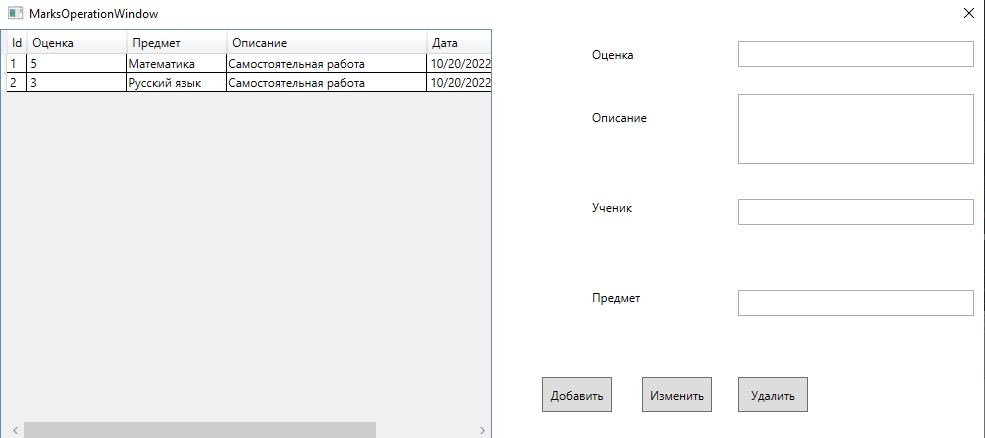


Рис. 27 – Окно операций с оценками

10) Для удаления или изменения поля, нужно выбрать его в таблице слева, его информация перейдет в соотвутсвующие поля справа. Можно либо удалить поле, либо отредактировать информацию и нажать кнопку “Изменить”.

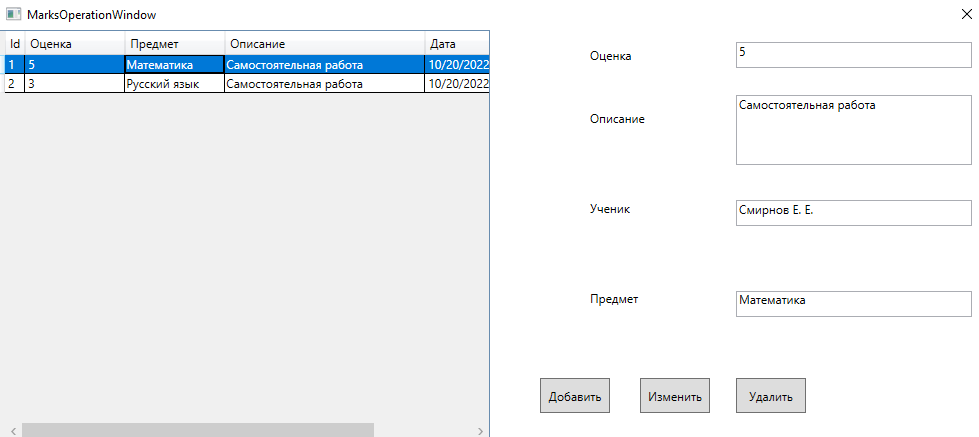


Рис. 28 – Окно операций с оценками

11) Для добавления уроков в расписание необходимо заполнить соответствующие поля и нажать на кнопку “Добавить”.

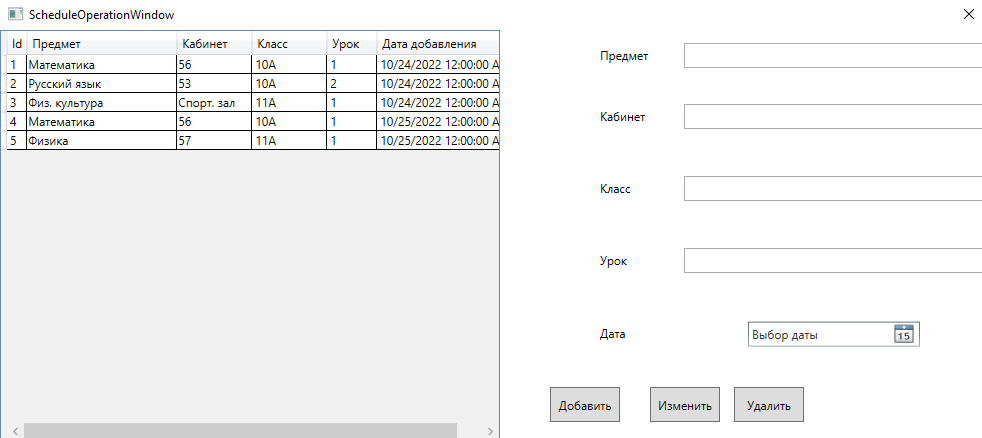


Рис. 29 – Окно операций с расписанием

12) Для удаления или изменения поля, нужно выбрать его в таблице слева, его информация перейдет в соотвутсвующие поля справа. Можно либо удалить поле, либо отредактировать информацию и нажать кнопку “Изменить”.

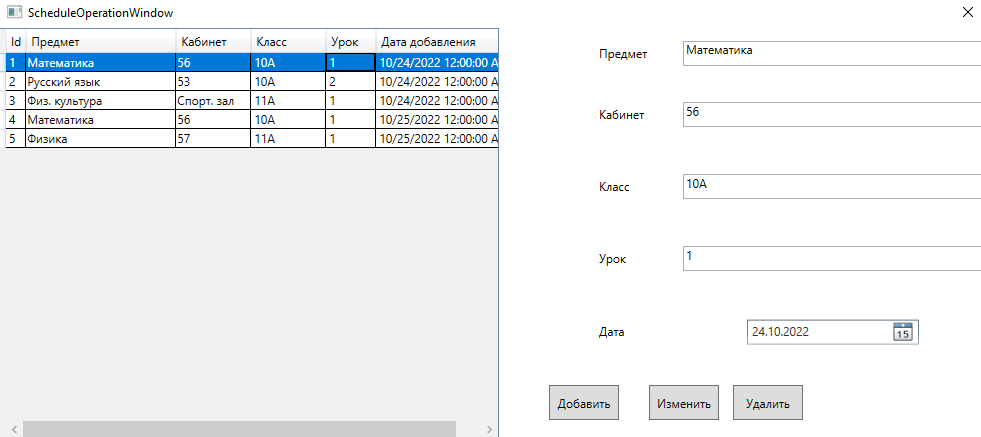


Рис. 30 – Окно операций с расписанием

13) Для добавления пользователей и их акаунтов необходимо заполнить соответствующие поля и нажать на кнопку “Добавить”. Здесь можно выдать или забрать права администратора.

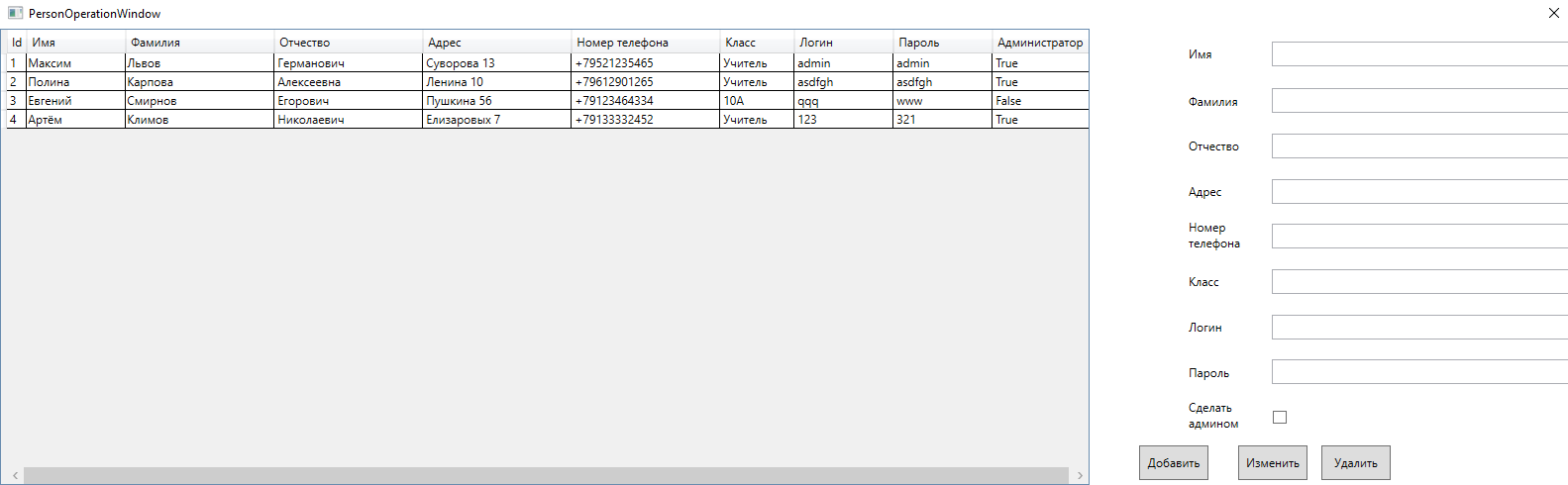


Рис. 31 - Окно операций с пользователями

14) Для удаления или изменения поля, нужно выбрать его в таблице слева, его информация перейдет в соотвутсвующие поля справа. Можно либо удалить поле, либо отредактировать информацию и нажать кнопку “Изменить”.

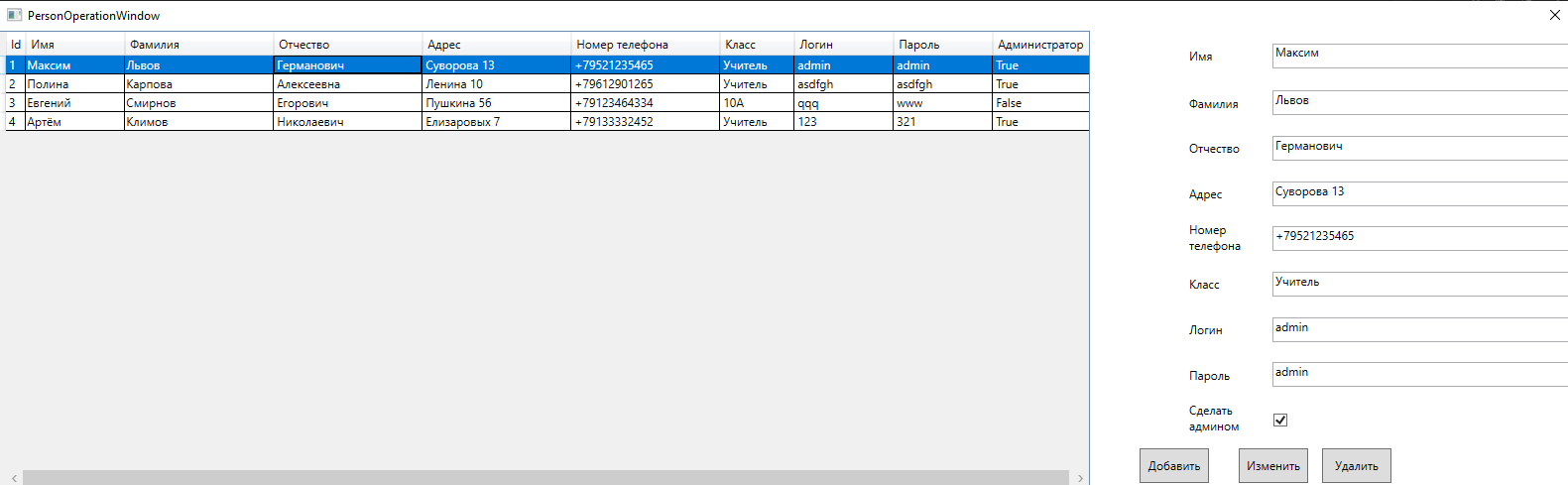


Рис. 32 – Окно операций с пользователями

15) Для добавления предметов необходимо заполнить соответствующие поля и нажать на кнопку “Добавить”.

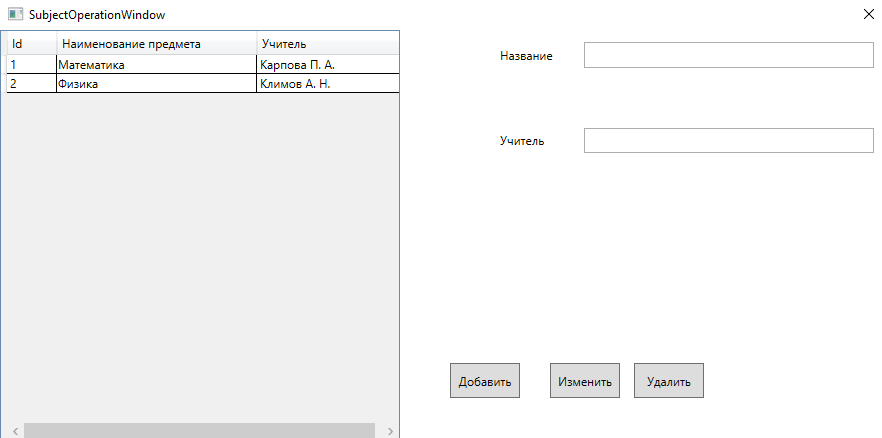


Рис. 33 – Окно операций с предметами

16) Для удаления или изменения поля, нужно выбрать его в таблице слева, его информация перейдет в соотвутсвующие поля справа. Можно либо удалить поле, либо отредактировать информацию и нажать кнопку “Изменить”.

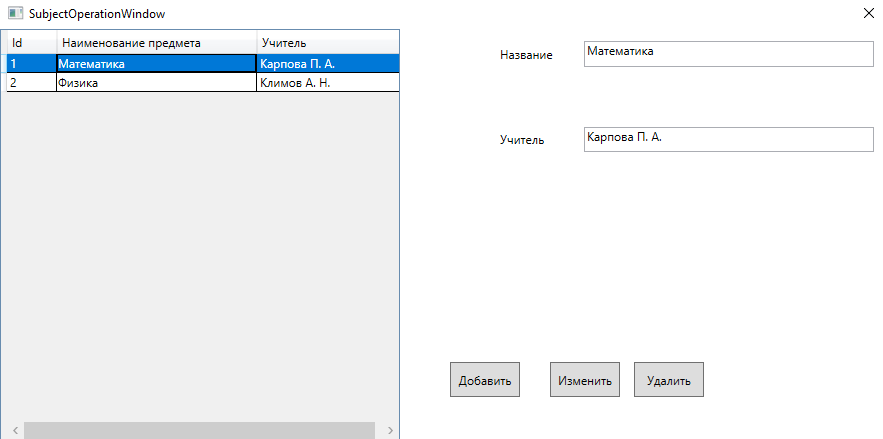


Рис. 34 – Окно операций с предметами