

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ»)

Строительно-политехнический колледж

(наименование структурного подразделения)

**Отчет**

о прохождении учебной практики профессионального модуля  
ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Название практики: Учебная практика Разработка модулей программного обеспечения для компьютерных систем

Обучающегося Клочковой Анастасии Александровны

(Ф.И.О.)

3 курса ИСП-2010 группы

Сроки прохождения практики:

с «22» апреля 2023 г. по «28» апреля 2023г.

в объеме 36 час.

Место прохождения практики ФГБОУ ВО «ВГТУ», СПК.

(наименование организации)

Итоговая оценка

отлично

Дата

28.04.2023

Руководитель по практической подготовке

от организации

(подпись, дата)

(И.О. Фамилия, должность)

Руководитель по практической

подготовке от ВГТУ

28.05.23 А.С. Дюдина

(подпись, дата)

(И.О. Фамилия, должность)

Воронеж 2023

Воронеж 2023

### ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

По учебной практике профессионального модуля ПМ.01 Разработка модулей программного обеспечения для компьютерных систем.

Название практики Учебная практика Разработка модулей программного обеспечения для компьютерных систем

№	Содержание заданий	Дата выполнения
1.	Разработка программного продукта по предметной области «Деканат»	22.04.2023
2.	Оформление отчёта, выводы.	28.04.2023

Задание принято к исполнению: Обучающийся \_\_\_\_\_  
(подпись)

Руководитель практики

от СПК

\_\_\_\_\_  
(подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия, должность)

## Содержание

Введение.....	4
1.1. Анализ предметной области .....	6
1.2. Разработка технического задания.....	7
1.3. Анализ требований к программному продукту .....	9
1.4. Определение основных технических решений .....	10
1.5. Проектирование программного продукта .....	11
1.6. Разработка программного продукта.....	13
1.7. Тестирование программного продукта .....	15
ЗАКЛЮЧЕНИЕ .....	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	28
Приложение А .....	29



## Введение

Одним из наиболее перспективных способов эффективного управления образовательным процессом является внедрение информационно-коммуникативных технологий в процесс менеджмента на базе международных стандартов.

Развитие сферы информационных технологий дало толчок к усовершенствованию всех сфер жизнедеятельности общества.

Достаточно актуальным является вопрос автоматизации учебного процесса, в частности, автоматизации процессов управления в высших учебных заведениях (вузах).

Цель автоматизации управления – получить удобные средства мониторинга, анализа и отображения основных процессов приема студентов, административного управления, подготовки и выпуска специалистов университета (института, академии).

Следует отметить, что одной из основных и трудоёмких в работе вуза является деятельность деканата. Работникам деканатов приходится выполнять огромный объем рутинной работы по учету контингента студентов, обеспечению учебного процесса, предоставлению информации в различные подразделения вуза. При этом всю информацию необходимо представлять в различных форматах. При этом необходимо осуществлять прогноз успеваемости студентов с целью эффективного контроля за процессом обучения и своевременного реагирования на возможные негативные результаты (низкие показатели абсолютной и качественной успеваемости).

Таким образом, необходимость внедрения информационной системы, автоматизирующей основные функции образовательного процесса, является актуальной. Прежде чем внедрять автоматизированную информационную систему (АИС) в деканате, как и в любом другом подразделении вуза, необходимо определить основные требования к её работе. Выявленные проблемы предметной области помогут определить направление автоматизации данной сферы деятельности, а разработка модели предметной области обеспечит успешную разработку и внедрение базы данных, как основной части АИС.

Необходимо выполнить задачи, которые исходят из цели работы:

1. Разработать техническое задание в соответствии со своей предметной областью и выполнить анализ функциональных и эксплуатационных требований.
2. Определить основные технические решения.
3. Спроектировать программный продукт, построив необходимые для этого диаграммы.
4. Разработать программный продукт.
5. Произвести отладку программы.
6. Провести оценку качества в соответствии со стандартами и техническим заданием.

Приложение, которое мы разработали, позволяет взаимодействовать с базой данных, а также предоставляет пользователю понятный интерфейс.

### 1.1. Анализ предметной области

Для успешного и эффективного взаимодействия студентов и преподавателей в ВУЗе необходимо правильно составленное расписание.

Одним из структурных подразделений ВУЗа является деканат.

Деканат – подразделение ВУЗа, осуществляющее контроль и организацию деятельности факультета.

В отделе по составлению расписания для студентов и учителей в распоряжении имеются: сведения о сформированных группах студентов, данные о штате преподавателей, информация об аудиториях и об изучаемых дисциплинах.

Основные задачи, которые необходимо решать в процессе управленческой деятельности деканата:

- разработать концептуальную и физическую модели базы данных хранения информации о студентах;
- обосновать выбор методов и моделей обработки и анализа данных по результатам обучения студентов;
- провести анализ и обосновать выбор существующих программных инструментариев для разработки систем хранения информации и осуществления эффективного документооборота;
- реализовать в виде конечного программного продукта систему «Деканат», которая обеспечит ввод, хранение и анализ информации о студентах и результатах их учебы.

Объектом исследования является процесс ввода, хранения и анализа информации о результатах обучения студентов.

Разрабатываемая система должна быть предназначена для автоматизации обработки, визуализации и анализа данных из БД, к которым относятся процессы, связанные с чтением и обновлением данных, добавлением и удалением данных.

Разрабатываемая система должна быть представлена в виде единого приложения. Разрабатываемая система должна отражать специфику предметной области и представлять собой единое целое.

Требования к надежности. В случае ошибок, вызванных некорректными действиями пользователя при работе, программа должна выдавать соответствующее сообщение и продолжать нормальную работу.

Создаваемое программное обеспечение должно обеспечить дружелюбный интерфейс пользователя – стандартное расположение компонентов управления (рекомендованное для данной операционной системы), дублирование команд, наличие всплывающих подсказок. При организации интерфейса необходимо использовать цвета в соответствии с правилами технической этики и текущей системной пользовательской палитры. Интерфейс пользователя должен обеспечивать эффективное взаимодействие пользователя с системой с минимальной необходимостью обращения к дополнительным справочным материалам.

## 1.2. Разработка технического задания

Название приложения: Деканат

Основные функции:

- Добавление, изменение и удаление данных о расписаниях, студентах и преподавателях;
- Поиск данных в таблице расписаний.

Требования к программному обеспечению:

- Приложение должно быть доступно на устройствах с операционной системой Windows;
- Приложение должно иметь понятный и простой интерфейс;
- Приложение должно обеспечивать безопасность и конфиденциальность данных.

Технические требования:

- Поддержка базы данных для хранения информации о расписании занятий;
- Использование библиотеки Entity Framework;
- Передача информации между клиентом и сервером должна быть защищенной.

Системные требования:

- Процессор: 4-ядерный или выше;

- ОЗУ: 4 Гб или выше;
- Свободное место на устройстве: не менее 5 Гб.

Тестирование:

- Обнаруженные ошибки и недоработки должны быть исправлены в кратчайшие сроки.

Поддержка:

- Поддержка приложения и обновления функций должны быть доступны для студентов в течение всего срока обучения.

Для реализации дополнительных функций базы может потребоваться введение дополнительных атрибутов, например, табельного номера и домашнего телефона студента, домашнего адреса и домашнего телефона преподавателя, должности преподавателя, рабочей программы, даты сдачи экзамена (зачета) и т.д.

О преподавателях есть следующая информация:

- Название кафедры;
- Должность;
- Стаж работы;
- ФИО;
- Оклад;
- Специальность;
- Табельный номер;
- Расписание.

Они обеспечивают проведение занятий по некоторым предметам.

Также курируют группу, о которой известно:

- Курс;
- Факультет;
- Направление;
- Количество студентов;
- Название;



- Номера телефонов студентов и куратора.

О студентах будет известна следующая информация:

- ФИО;
- Номер телефона;
- Группа.

Они ходят на пары по разным предметам:

- Название;
- Вид пар (Лекция, семинар или лабораторная);
- День недели;
- Время пары;
- Номер аудитории;
- Группа;
- Преподаватель.

### 1.3. Анализ требований к программному продукту

Функциональные требования пользователя:

- автоматизация работы специалистов отдела расписания;
- возможность просмотра доступной информации для студентов и преподавателей.

Входные данные:

Входными данными при работе с программным продуктом должны быть данные о преподавателях, группах студентов, типах занятий, аудиториях и времени, проводимых занятий. Входной язык приложения - русский.

Выходные данные:

Выходными данными при работе программы являются данные (расписание), выводимые на экран пользователя. Язык выходных данных в приложении - русский.

Требования к интерфейсу:

- Понятность и логичность;
- Функциональность.

Требования к надежности:

- контроль вводимой информации, т.е. возможность отслеживания ошибок, допускаемых пользователем, и последующей реакции программы на них.

#### 1.4. Определение основных технических решений

В качестве среды разработки для написания программы был выбран Microsoft Visual Studio. Это интегрированная среда разработки (IDE), которая включает в себя редактор кода, отладчик и тестовую среду. Visual Studio поддерживает широкий спектр языков программирования, в том числе C#.

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

C# — это современный объектно-ориентированный язык программирования, который был разработан компанией Microsoft. Он имеет динамическую типизацию, а также возможности для создания GUI (графических пользовательских интерфейсов). Кроме того, в C# реализованы средства безопасности в памяти и управление игроками.

Состав функций программного продукта:

- Функции чтения/добавления/удаления/изменения данных о расписаниях, студентах и преподавателях в базе данных.
- Функция поисковой системы для вкладки «Расписание».
- Функция автозаполнения полей ввода по номеру записи для всех вкладок.
- Функция обновления таблицы базы данных.

Режимы функционирования:

- Режим добавления новых данных. В этом режиме можно добавлять новых преподавателей, студентов, а также информацию о занятиях, которые нужно распределить в расписание.
- Режим просмотра расписания. В этом режиме пользователь может просмотреть готовое расписание.
- Режим изменения расписания. Если возникла необходимость внести изменения в готовое расписание, можно выбрать этот режим и внести нужные корректировки в расписание занятий.

### 1.5. Проектирование программного продукта

Для программы создали 4 диаграммы: Сущность-связь, Диаграмма потоков данных, Функциональная модель бизнес-процесса второго уровня. Построенные диаграммы представлены на рисунках 1.1-1.4.

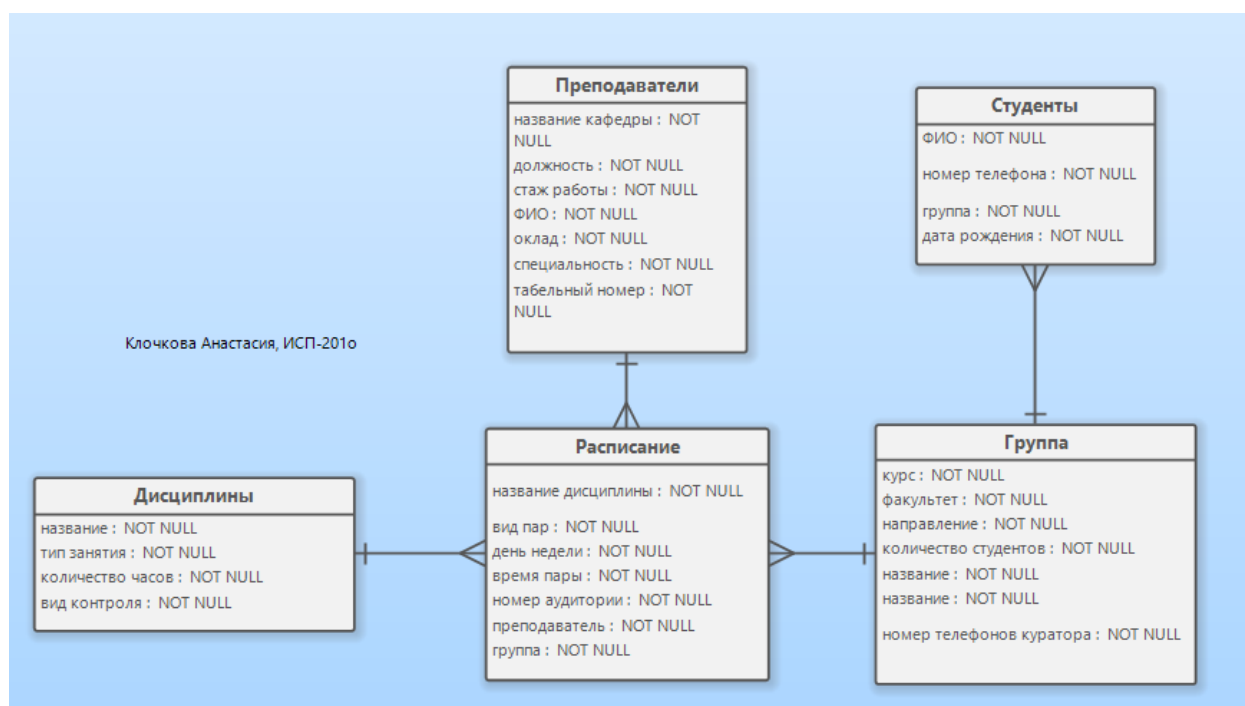


Рисунок 1.1 – Диаграмма сущность-связь



Рисунок 1.2 – Диаграмма потоков данных

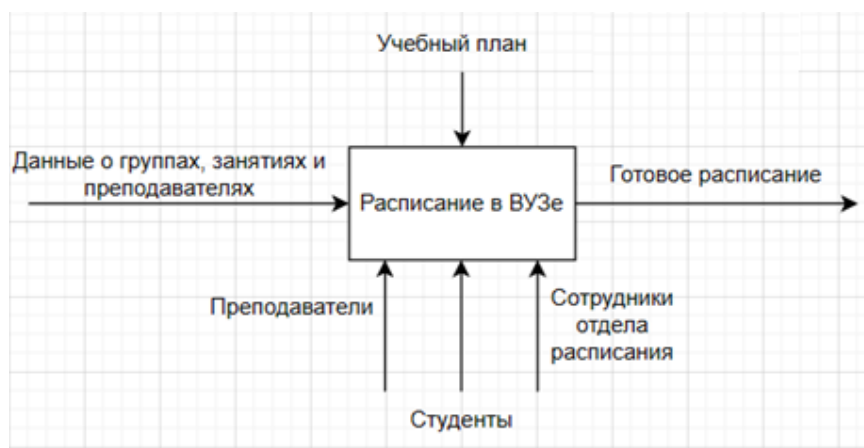


Рисунок 1.3 – Диаграмма IDEF0

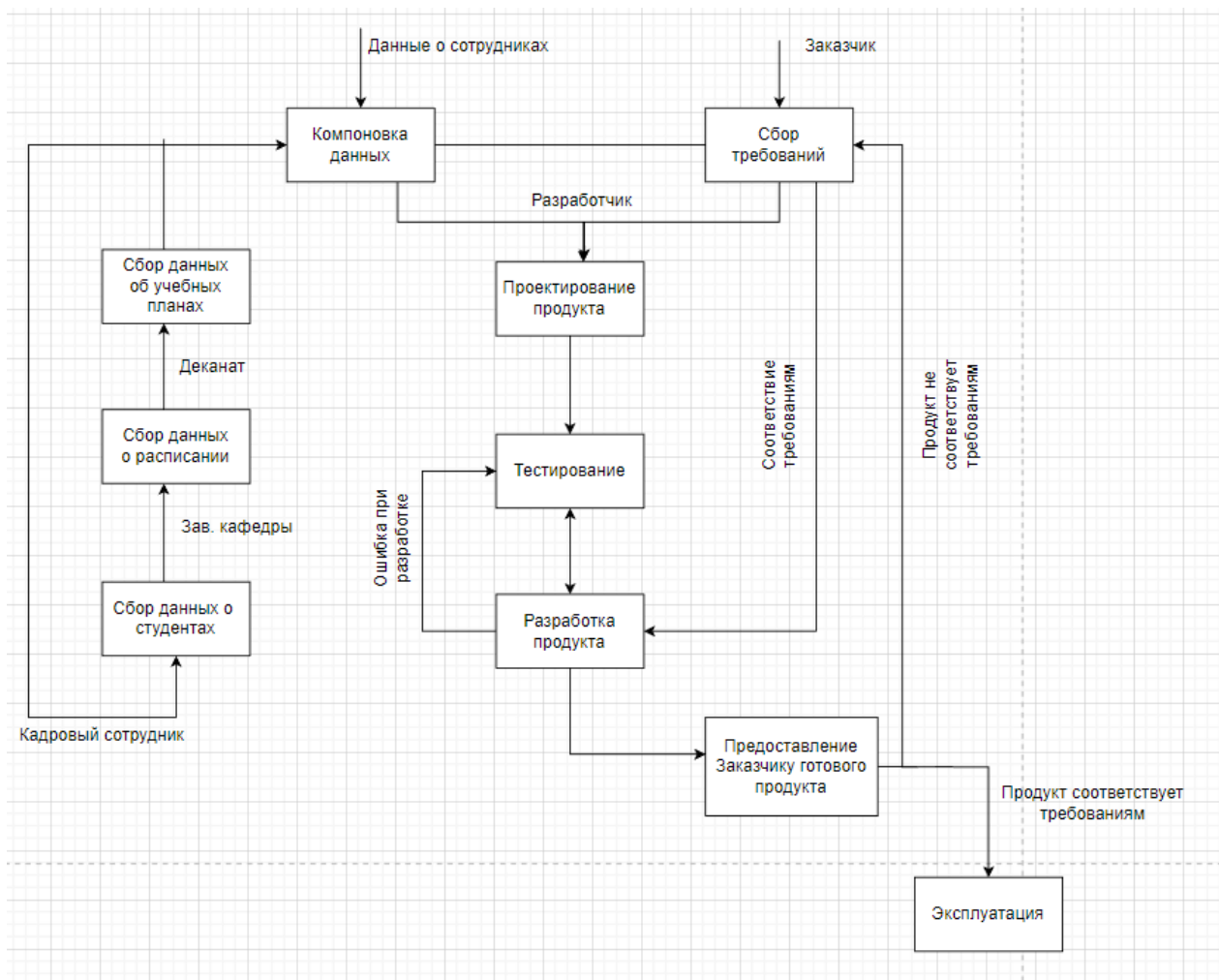


Рисунок 1.4 – Функциональная модель бизнес-процесса второго уровня

### 1.6. Разработка программного продукта

Данная программа разработана для составления расписания в ВУЗе. Она состоит из 3 вкладок:

- Расписание
- Преподаватели
- Студенты

В программе реализовано всего 7 классов (Timetable, Students, Teacher, Groups, Specialist, Faculties, Discipline). Классы используются для связи программы с хранилищем данных. Они позволяют упростить доступ к базе данных и сделать его более безопасным и надежным. Все эти классы содержат в себе поля, которые равны атрибутам сущностей в базе

Листинг кода программы представлен в приложении А.

Конструктор программы представлен на рисунках 1.5-1.7:

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
174	БГРАД-183	Красномызник ...	ПБ-192	Право	Лекция	Четвер
9	ПБ-201с	Сапогов Ричард...	ПБ-193	БГРАД-181	Практика	Вторни
10	БГРАД-182	Катамаранов И...	ПБ-192	ПБ-193	Практика	Вторни

Рисунок 1.5 – Конструктор программы

№	Специальность	Кафедра	Опыт_работы	ФИО	Зарплата
3	Доцент	Кафедра филос...	4 года	Крысова Иннес...	11000
4	Преподаватель	Кафедра литера...	20 лет	Красномызник ...	8000
6	Доцент	Кафедра физики	9 лет	Шершанский Як...	65000
7	Преподаватель	Кафедра изо	8 лет	Вишневский Гв...	5000
8	Преподаватель	Кафедра филос...	3 года	Восьмиглазова ...	33000
9	Доцент	Кафедра высок...	5 лет	Сапогов Ричард...	65000
10	Преподаватель	Кафедра Физку...	11 лет	Катамаранов И...	13000
12	Доцент	Кафедра филос...	13 лет	Петров Петр Ил...	13000

Рисунок 1.6 – Конструктор программы



	id	fio	phone_number	groupss	date_of_birth
▶	1	Абрамов Алекс...	+78956142155	ПБ-201	20.02.2001
	3	Кишмишева Алё...	+79456458256	ПБ-201	12.04.2004
	4	Тазиков Джама...	+79457988922	ПБ-203	28.09.2003
	5	Цукерник Кондр...	+79456458312	ПБ-201	27.01.2000
	6	Клёнова Мелис...	+79468558922	ПБ-204	25.07.2005
	8	Сильвесторов П...	+79456461822	ПБ-204	14.05.2002
	9	Шизоид Сергей ...	+79456458874	ПБ-203	21.08.1999
	10	Попловский Кш...	+79456458922	ПБ-201	12.06.2003
*					

Рисунок 1.7 – Конструктор программы

## 1.7. Тестирование программного продукта

Мы разработали программный продукт, произвели отладку и тестирование. Результаты тестирования представлены на рисунках 1.12-1.19.

**Для начала работы в приложении:**

1. Запустить программу.
2. Начать вводить данные, как описано ниже (рисунок 1.12).

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
174	бГРАД-183	Красномызник ...	ПБ-192	Право	Лекция	Четвер
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторни
10	бГРАД-182	Катамаранов И...	ПБ-192	ПБ-193	Практика	Вторни

Рисунок 1.12 – АИС «Деканат»

Расписание создаётся по кнопке «Добавить». Вся справочная информация выбирается из выпадающего списка (рисунок 1.13).

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
174	бГРАД-183	Красномызник ...	ПБ-192	Право	Лекция	Четвер
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторни
10	бГРАД-182	Катамаранов И...	ПБ-192	ПБ-193	Практика	Вторни

Рисунок 1.13 – АИС «Деканат»

Предусмотрена проверка на ввод всех необходимых данных, как показано на рисунке 1.14.

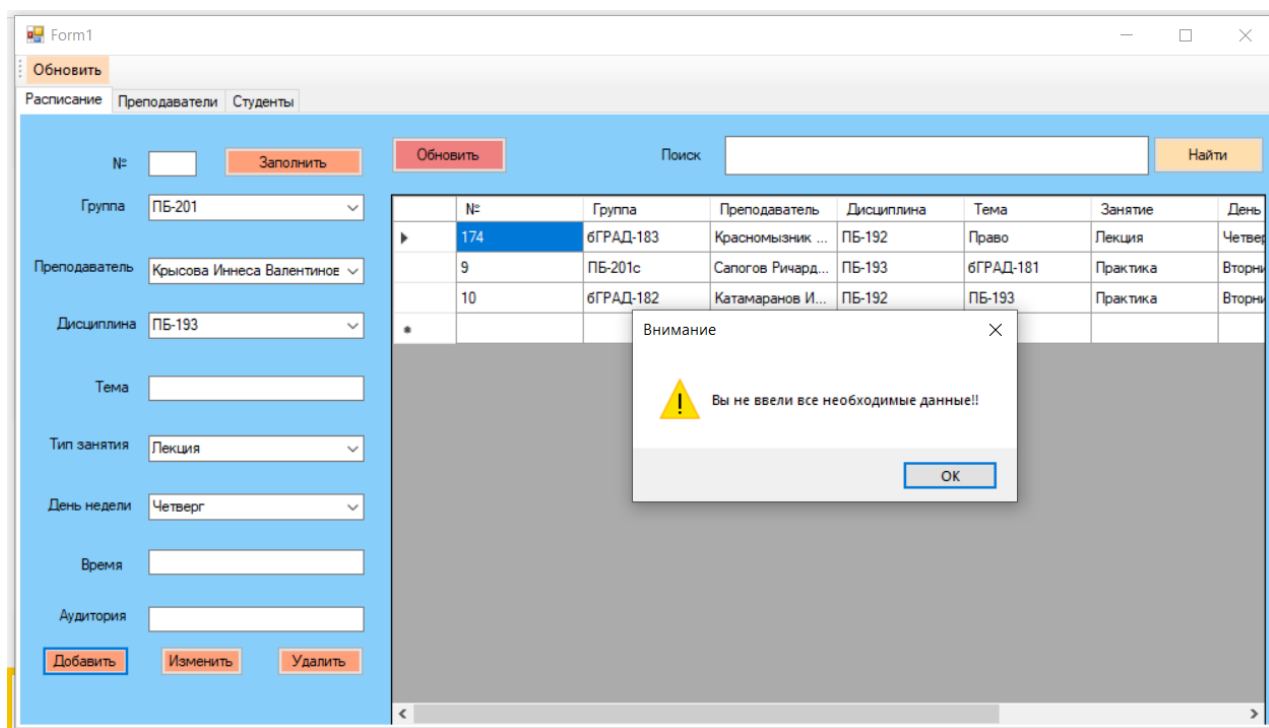


Рисунок 1.14 – АИС «Деканат»

При добавлении записи вылетает окно «Информация добавлена» и автоматически обновляет таблицу (рисунок 1.15).

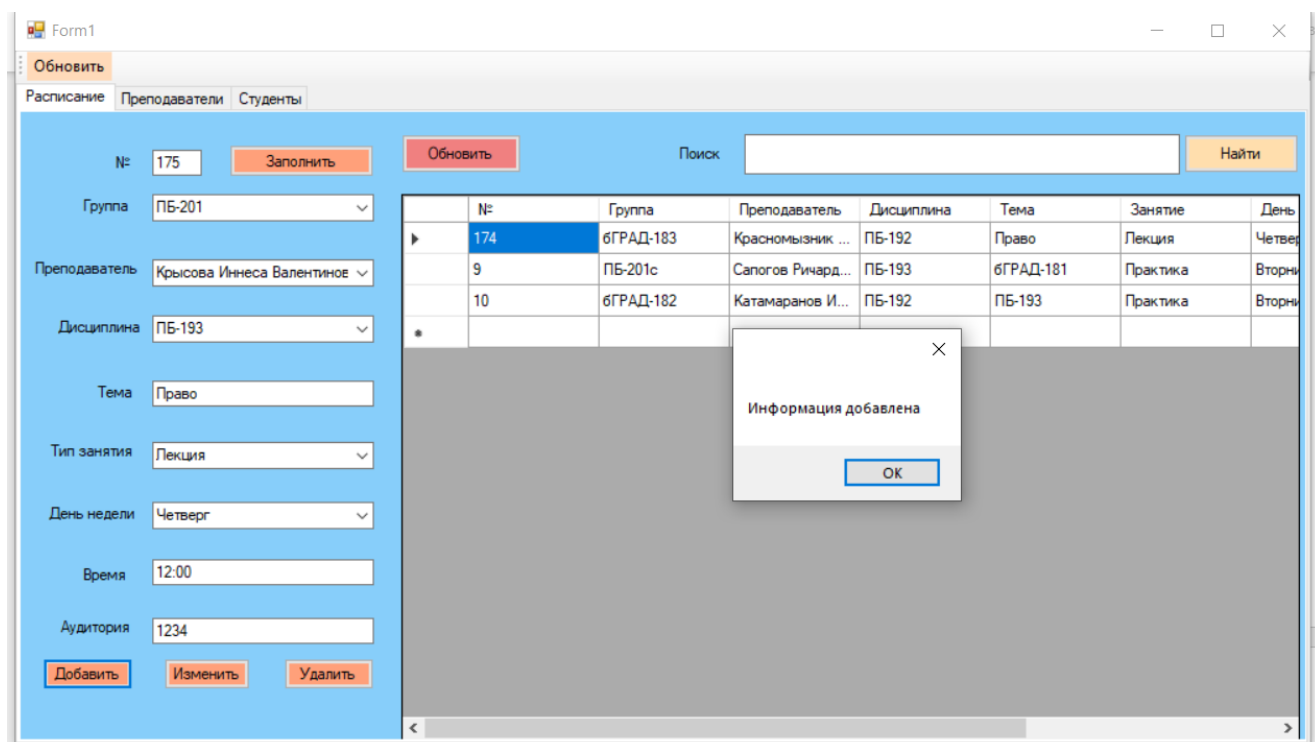


Рисунок 1.15 – АИС «Деканат»

Предусмотрена функция Автозаполнения полей ввода по номеру (рисунок 1.16).

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
175	ПБ-201	Восьмиглазова ...	ПБ-193	Право	Лекция	Четверг
174	бГРАД-183	Красномызник ...	ПБ-192	Право	Лекция	Четверг
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторник
10	бГРАД-182	Катамаранов И...	ПБ-192	ПБ-193	Практика	Вторник

Рисунок 1.16 – АИС «Деканат»

Если не заполнить необходимое поле для ввода, то вылетает окно «Необходимо заполнить данные».

Расписание изменяется по кнопке «Изменить». Вся справочная информация выбирается из выпадающего списка (рисунок 1.17).

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
175	ПБ-201	Восьмиглазова ...	ПБ-193	Право	Лекция	Четверг
174	бГРАД-183	Красномызник ...	ПБ-192	Право	Лекция	Четверг
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторник
10	бГРАД-182	Катамаранов И...	ПБ-192	ПБ-193	Практика	Вторник

Рисунок 1.17 – АИС «Деканат»

При изменении записи вылетает окно «Информация изменена» и автоматически обновляет таблицу (рисунок 1.18).

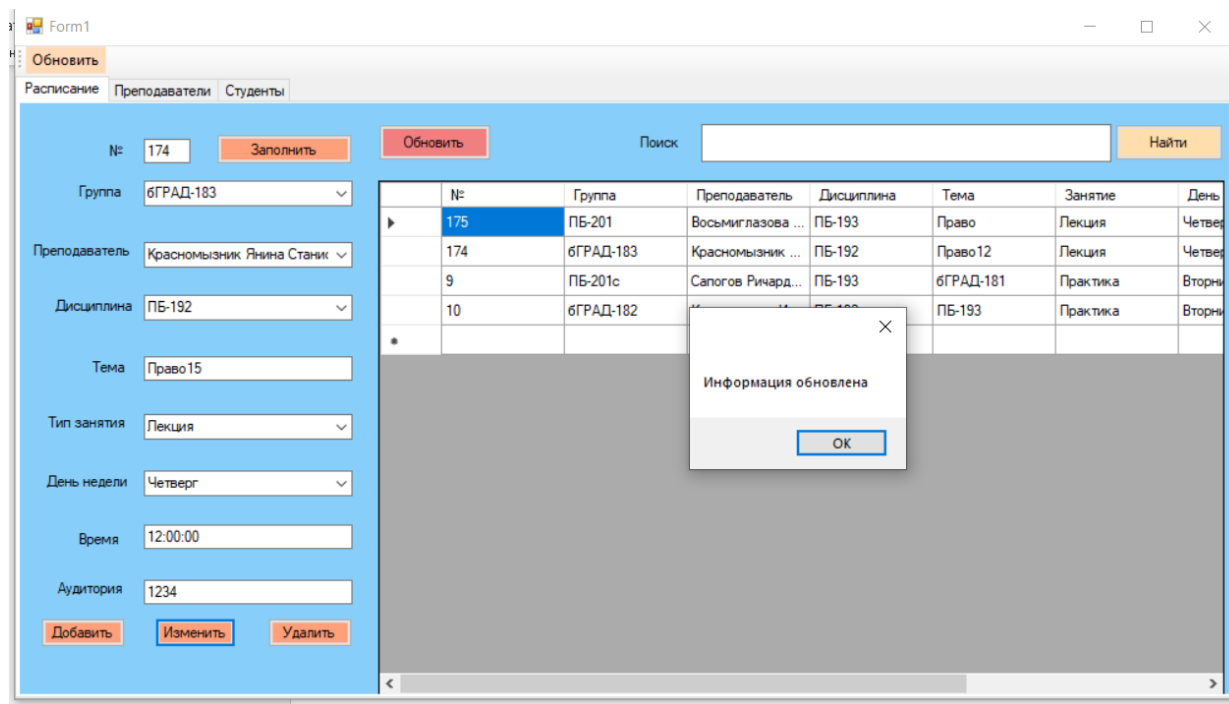


Рисунок 1.18 – АИС «Деканат»

Расписание удаляется по кнопке «Удалить». В поле «№» вводится номер записи, которую необходимо удалить.

При удалении записи вылетает окно «Информация удалена» и автоматически обновляет таблицу (рисунок 1.19).

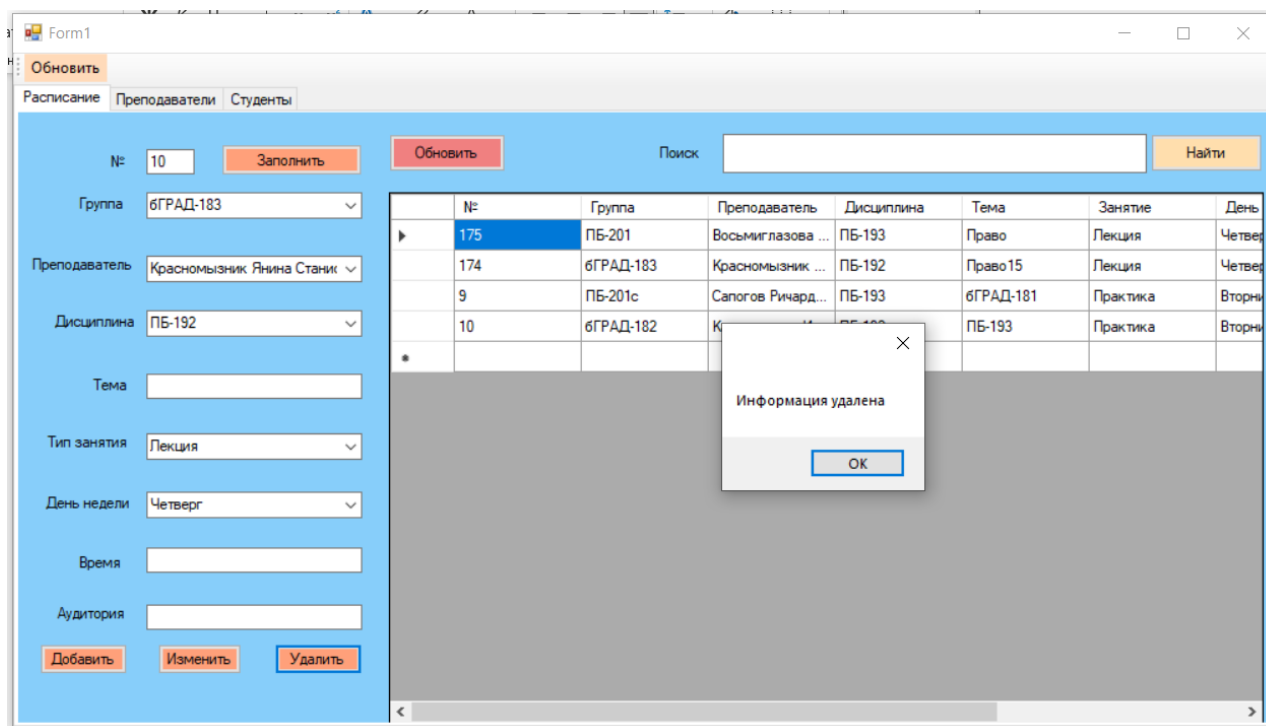


Рисунок 1.19 – АИС «Деканат»

Также предусмотрена проверка на незаполненные поля (рисунок 1.20).

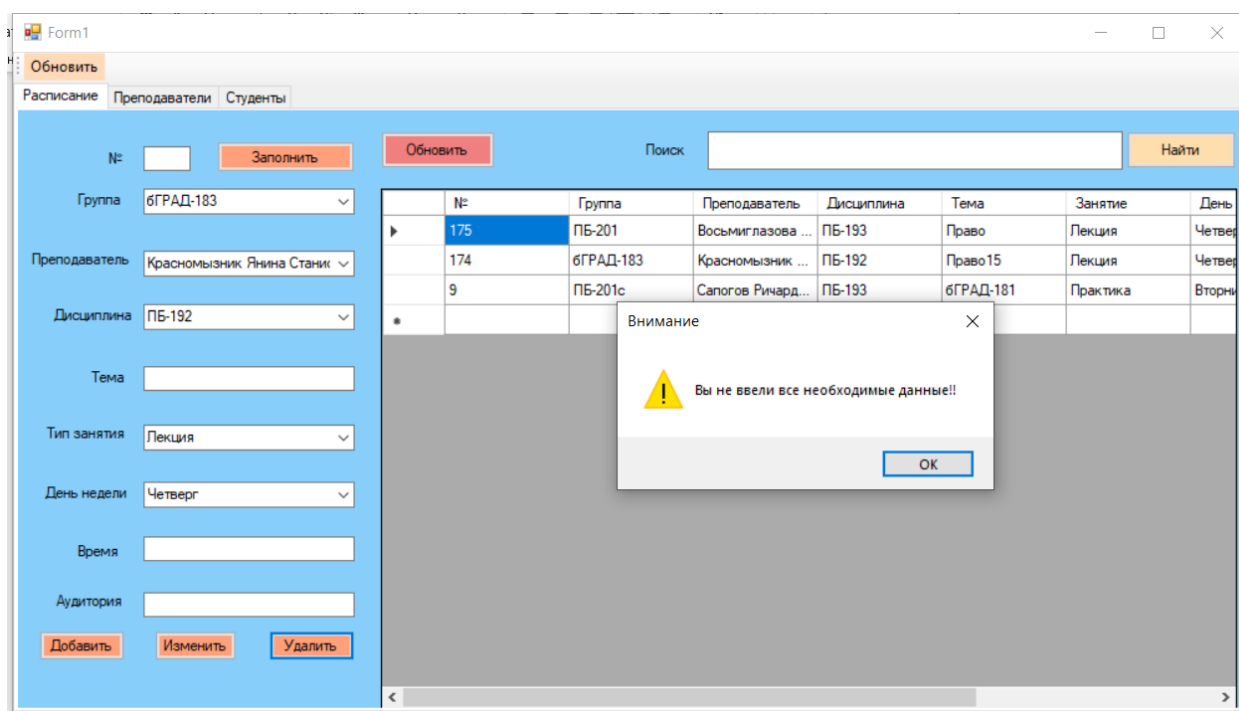


Рисунок 1.20 – АИС «Деканат»

Предусмотрен фильтр данных через поисковую структуру (рисунок 1.21).



Form1

Обновить

Расписание Преподаватели Студенты

№  Заполнить Обновить Поиск  Найти

Группа

Преподаватель

Дисциплина

Тема

Тип занятия

День недели

Время

Аудитория

Добавить Изменить Удалить

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторни
*						

Рисунок 1.21 – АИС «Деканат»

Кнопка «Обновить» не только обновляет данные, но и сбрасывает фильтр (рисунок 1.22).

Form1

Обновить

Расписание Преподаватели Студенты

№  Заполнить Обновить Поиск  Найти

Группа

Преподаватель

Дисциплина

Тема

Тип занятия

День недели

Время

Аудитория

Добавить Изменить Удалить

№	Группа	Преподаватель	Дисциплина	Тема	Занятие	День
175	ПБ-201	Восьмиглазова ...	ПБ-193	Право	Лекция	Четвер
174	бГРАД-183	Красномызник ...	ПБ-192	Право15	Лекция	Четвер
9	ПБ-201с	Сапогов Ричард...	ПБ-193	бГРАД-181	Практика	Вторни
*						

Рисунок 1.22 – АИС «Деканат»

На вкладке «Преподаватель» для редактирования данных также есть кнопки «Добавить», «Изменить», «Удалить» (рисунок 1.23).

№	Специальность	Кафедра	Опыт работы	ФИО	Зарплата
3	Доцент	Кафедра филос...	4 года	Крысова Иннес...	11000
4	Преподаватель	Кафедра литера...	20 лет	Красномызник ...	8000
6	Доцент	Кафедра физики	9 лет	Шершанский Як...	65000
7	Преподаватель	Кафедра изо	8 лет	Вишневыский Гв...	5000
8	Преподаватель	Кафедра филос...	3 года	Восьмиглазова ...	33000
9	Доцент	Кафедра высок...	5 лет	Сапогов Ричард...	65000
10	Преподаватель	Кафедра Физку...	11 лет	Катамаранов И...	13000
12	Доцент	Кафедра филос...	13 лет	Петров Петр Ил...	13000

Рисунок 1.23 – АИС «Деканат»

Проверка полей на пустоту при добавлении записи (рисунок 1.24).

Внимание

Вы не ввели все необходимые данные!!

OK

Рисунок 1.24 – АИС «Деканат»

При добавлении записи вылетает окно и данные автоматически обновляются в таблице (рисунок 1.25).

Form1

Обновить

Расписание Преподаватели Студенты

Обновить

№: 13 Заполнить

№ специальности: Доцент

Опыт работы: 2 года

ФИО: Иванов Иван Иванович

Зарплата (в руб.): 15000

Персональный номер: 7896

Добавить Изменить Удалить

№	Специальность	Кафедра	Опыт_работы	ФИО	Зарплата
3	Доцент	Кафедра филос...	4 года	Крысова Иннес...	11000
4	Преподаватель	Кафедра литера...	20 лет	Красномызник ...	8000
6	Доцент	Кафедра физики	9 лет	Шершанский Як...	65000
7	Преподаватель	Кафедра изо	8 лет	Вишневский Гв...	5000
8	Преподаватель	Кафедра физик...	3 года	Восьмиглазова ...	33000
9	Доцент	Кафедра высок...	5 лет	Сапогов Ричард...	65000
10	Преподаватель	Кафедра Физку...	11 лет	Катамаранов И...	13000
12	Доцент	Кафедра филос...	13 лет	Петров Петр Ил...	13000
13	Доцент	Кафедра филос...	2 года	Иванов Иван И...	15000

Информация добавлена

OK

Рисунок 1.25 – АИС «Деканат»

Кнопки «Изменить» и «Удалить» работают аналогично.

Также предусмотрена функция автозаполнения всех полей по номеру (рисунок 1.26).

Form1

Обновить

Расписание Преподаватели Студенты

Обновить

№: 6 Заполнить

№ специальности: Доцент

Опыт работы: 9 лет

ФИО: Шершанский Яков Осипович

Зарплата (в руб.): 65000

Персональный номер: 9376

Добавить Изменить Удалить

№	Специальность	Кафедра	Опыт_работы	ФИО	Зарплата
3	Доцент	Кафедра филос...	4 года	Крысова Иннес...	11000
4	Преподаватель	Кафедра литера...	20 лет	Красномызник ...	8000
6	Доцент	Кафедра физики	9 лет	Шершанский Як...	65000
7	Преподаватель	Кафедра изо	8 лет	Вишневский Гв...	5000
8	Преподаватель	Кафедра филос...	3 года	Восьмиглазова ...	33000
9	Доцент	Кафедра высок...	5 лет	Сапогов Ричард...	65000
10	Преподаватель	Кафедра Физку...	11 лет	Катамаранов И...	13000
12	Доцент	Кафедра филос...	13 лет	Петров Петр Ил...	13000
13	Доцент	Кафедра филос...	2 года	Иванов Иван И...	15000

Рисунок 1.26 – АИС «Деканат»

Также есть кнопка обновления данных.

Вкладка «Студенты» спроектирована схожим образом с вкладкой «Преподаватели».

Для редактирования данных используются кнопки «Добавить», «Изменить», «Удалить» (рисунок 1.27).

	id	fio	phone_number	groupss	date_of_bith
▶	1	Абрамов Алекс...	+78956142155	ПБ-201	20.02.2001
	3	Кишмишева Але...	+79456458256	ПБ-201	12.04.2004
	4	Тазиков Джама...	+79457988922	ПБ-203	28.09.2003
	5	Цукерник Кондр...	+79456458312	ПБ-201	27.01.2000
	6	Клёнова Мелис...	+79468558922	ПБ-204	25.07.2005
	8	Сильвесторов П...	+79456461822	ПБ-204	14.05.2002
	9	Шизоид Сергей ...	+79456458874	ПБ-203	21.08.1999
	10	Попловский Кш...	+79456458922	ПБ-201	12.06.2003
*					

Рисунок 1.27 – АИС «Деканат»

При добавлении, изменении, удалении записи вылетает окно и данные автоматически обновляются в таблице (рисунок 1.28).

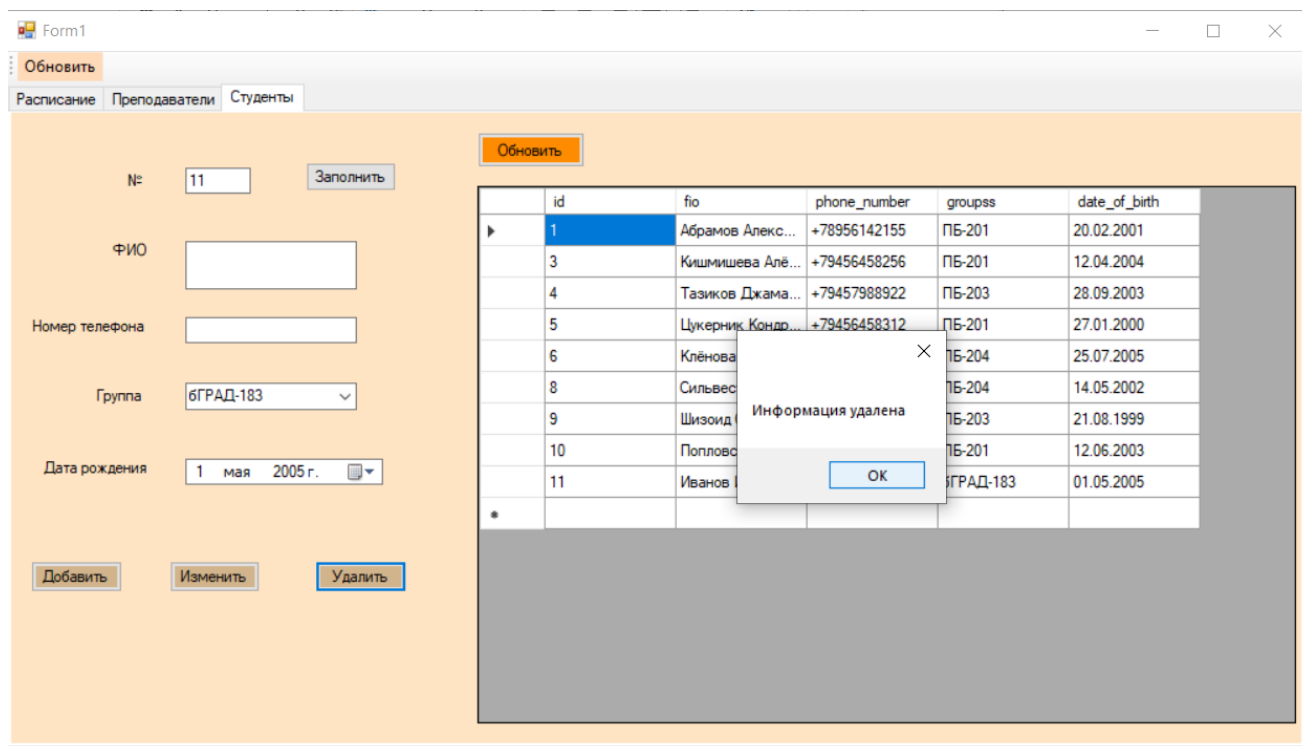


Рисунок 1.28 – АИС «Деканат»

При наличии пустых полей вылетает окно (рисунок 1.29).

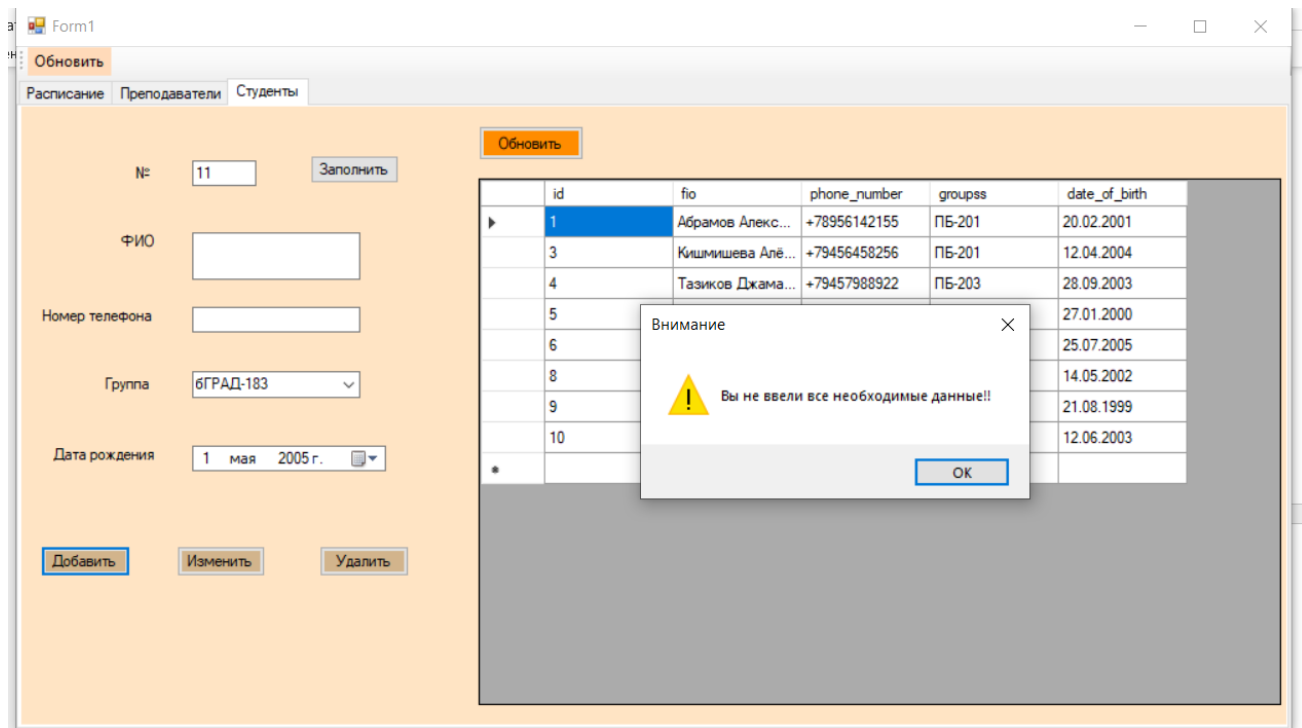


Рисунок 1.29 – АИС «Деканат»

Имеется функция автозаполнения по номеру (кнопка «Заполнить») (рисунок 1.30).

The screenshot shows a software window titled 'Form1'. It has three tabs: 'Расписание' (Schedule), 'Преподаватели' (Teachers), and 'Студенты' (Students), with 'Студенты' being the active tab. The form on the left contains several input fields: '№' (Number) with the value '10' and a 'Заполнить' (Fill) button; 'ФИО' (Full Name) with the value 'Попловский Кшиштоф Борисович'; 'Номер телефона' (Phone Number) with the value '+79456458922'; 'Группа' (Group) with a dropdown menu showing 'ПБ-201'; and 'Дата рождения' (Date of Birth) with a date picker set to '12 июня 2003 г.'. Below these fields are three buttons: 'Добавить' (Add), 'Изменить' (Edit), and 'Удалить' (Delete). An 'Обновить' (Update) button is located above the table. The table on the right has columns: 'id', 'fio', 'phone\_number', 'groupss', and 'date\_of\_birth'. It contains 10 rows of student data, with the first row highlighted in blue.

	id	fio	phone_number	groupss	date_of_birth
▶	1	Абрамов Алекс...	+78956142155	ПБ-201	20.02.2001
	3	Кишмишева Алё...	+79456458256	ПБ-201	12.04.2004
	4	Тазиков Джама...	+79457988922	ПБ-203	28.09.2003
	5	Цукерник Кондр...	+79456458312	ПБ-201	27.01.2000
	6	Клёнова Мелис...	+79468558922	ПБ-204	25.07.2005
	8	Сильвесторов П...	+79456461822	ПБ-204	14.05.2002
	9	Шизоид Сергей ...	+79456458874	ПБ-203	21.08.1999
	10	Попловский Кш...	+79456458922	ПБ-201	12.06.2003
*					

Рисунок 1.30 – АИС «Деканат»

Также есть кнопка для обновления данных.



## ЗАКЛЮЧЕНИЕ

В данной учебной практике было разработано приложение, которое взаимодействует с базой данных.

Изначально мы написали техническое задание, провели анализ функциональных требований и выделили основные технические решения. В приложении Draw io была разработана диаграмма сущность-связь и диаграмма потоков данных. Далее, разработали программный продукт, провели отладку и тестирование. Затем мы произвели оценку качества программного продукта.

Была установлена СУБД – PostgreSQL. На основе реляционной модели данных, полученной ранее, была создана база данных.

Благодаря созданию программы, заметно снизится количество ошибок, совершаемых сотрудниками отдела расписания в процессе работы и повысится производительность. Пользователи же в свою очередь получают простой и удобный интерфейс, а также возможность быстро получить необходимую информацию.

Разработанная АИС «Деканат» решает задачи тактического уровня управления. Данная система позволяет повысить эффективность работы деканатов высших учебных заведений. Для успешного и эффективного функционирования вуза в целом необходимо комплексное внедрение информационно-коммуникационных технологий, охватывающее все без исключения функциональные подсистемы учебного заведения (такие как деканаты, библиотека, приёмная комиссия, бухгалтерия, кафедры и др.)

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Стасышин, В.М. Базы данных: технологии доступа: учебное пособие для среднего профессионального образования/ В.М.Стасышин, Т.Л.Стасышина. — 2-е изд., испр. и доп.— Москва: Издательство Юрайт, 2020. — 164 с. — Режим доступа: <https://urait.ru/bcode/455863> -ЭБС Юрайт.
2. Туманов В.Е. Основы проектирования реляционных баз данных [Электронный ресурс]/ Туманов В.Е.— Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 502 с.— Режим доступа: <http://www.iprbookshop.ru/52221.html>.— ЭБС «IPRbooks»
3. Кулыгин О.П. Администрирование баз данных. СУБД MS SQL Server [Электронный ресурс]: учебное пособие/ Кулыгин О.П. - Электрон.текстовые данные. - М.: Московский финансово-промышленный университет «Синергия», 2012. - 232 с. - Режим доступа:<http://www.iprbookshop.ru/17009>
4. <https://www.sql.ru/>- SQL Сайт, посвященный SQL, программированию, базам данных, разработке информационных систем

## Приложение А

```
using Microsoft.EntityFrameworkCore;
using Microsoft.VisualBasic.ApplicationServices;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;

namespace practika12
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            BindingList<Auditorium> data1 = new BindingList<Auditorium>();
            BindingList<Groups_of_students> data2 = new
BindingList<Groups_of_students>();
            BindingList<Item> data3 = new BindingList<Item>();
            BindingList<Schedule> data4 = new BindingList<Schedule>();
            BindingList<Staff_of_teachers> data5 = new
BindingList<Staff_of_teachers>();
            BindingList<Type_of_classes_conducted> data6 = new
BindingList<Type_of_classes_conducted>();
            BindingList<Venue> data7 = new BindingList<Venue>();
            public void DGRefresh1()
            {
                data1.Clear();
                using (ApplicationContext db = new ApplicationContext())
                {
                    var infolist = db.auditorium.FromSqlRaw("SELECT * FROM
auditorium").ToList();
                    for (int i = 0; i < infolist.Count(); i++)
                    {
                        data1.Add(infolist[i]);
                        dataGridView1.DataSource = data1;
                    }
                }
            }
            public void DGRefresh2()
            {
                data2.Clear();
                using (ApplicationContext db = new ApplicationContext())
                {
                    var infolist = db.groups_of_students.FromSqlRaw("SELECT *
FROM groups_of_students").ToList();
                    for (int i = 0; i < infolist.Count(); i++)
                    {
                        data2.Add(infolist[i]);
                        dataGridView2.DataSource = data2;
                    }
                }
            }
        }
    }
}
```

```

        }
    }
}
public void DGRefresh3()
{
    data3.Clear();
    using (ApplicationContext db = new ApplicationContext())
    {
        var infolist = db.item.FromSqlRaw("SELECT * FROM
item").ToList();
        for (int i = 0; i < infolist.Count(); i++)
        {
            data3.Add(infolist[i]);
            dataGridView3.DataSource = data3;
        }
    }
}
public void DGRefresh4()
{
    data4.Clear();
    using (ApplicationContext db = new ApplicationContext())
    {
        var infolist = db.schedule.FromSqlRaw("SELECT * FROM
schedule").ToList();
        for (int i = 0; i < infolist.Count(); i++)
        {
            data4.Add(infolist[i]);
            dataGridView4.DataSource = data4;
        }
    }
}
public void DGRefresh5()
{
    data5.Clear();
    using (ApplicationContext db = new ApplicationContext())
    {
        var infolist = db.staff_of_teachers.FromSqlRaw("SELECT * FROM
staff_of_teachers").ToList();
        for (int i = 0; i < infolist.Count(); i++)
        {
            data5.Add(infolist[i]);
            dataGridView5.DataSource = data5;
        }
    }
}
public void DGRefresh6()
{
    data6.Clear();
    using (ApplicationContext db = new ApplicationContext())
    {

```

```

        var infolist =
db.type_of_classes_conducted.FromSqlRaw("SELECT * FROM
type_of_classes_conducted").ToList();
        for (int i = 0; i < infolist.Count(); i++)
        {
            data6.Add(infolist[i]);
            dataGridView6.DataSource = data6;
        }
    }
}
public void DGRefresh7()
{
    data7.Clear();
    using (ApplicationContext db = new ApplicationContext())
    {
        var infolist = db.venue.FromSqlRaw("SELECT * FROM
venue").ToList();
        for (int i = 0; i < infolist.Count(); i++)
        {
            data7.Add(infolist[i]);
            dataGridView7.DataSource = data7;
        }
    }
}
private void Form1_Load(object sender, EventArgs e)
{
    dataGridView1.DataSource = data1;
    dataGridView2.DataSource = data2;
    dataGridView3.DataSource = data3;
    dataGridView4.DataSource = data4;
    dataGridView5.DataSource = data5;
    dataGridView6.DataSource = data6;
    dataGridView7.DataSource = data7;

    DGRefresh1();
}

private void tabControl1_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (tabControl1.SelectedIndex == 0)
    {
        DGRefresh1();
    }

    if (tabControl1.SelectedIndex == 1)
    {
        DGRefresh2();
    }

    if (tabControl1.SelectedIndex == 2)

```

```

        {
            DGRefresh3();
        }

        if (tabControl1.SelectedIndex == 3)
        {
            DGRefresh4();
        }

        if (tabControl1.SelectedIndex == 4)
        {
            DGRefresh5();
        }

        if (tabControl1.SelectedIndex == 5)
        {
            DGRefresh6();
        }

        if (tabControl1.SelectedIndex == 6)
        {
            DGRefresh7();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            using (ApplicationContext db = new ApplicationContext())
            {
                Auditorium a = new
Auditorium(Convert.ToInt32(textBox1.Text), Convert.ToInt32(textBox2.Text),
Convert.ToInt32(textBox3.Text), Convert.ToInt32(textBox4.Text));

                db.auditorium.Add(a);
                db.SaveChanges();
                textBox1.Text = "";
                textBox2.Text = "";
                textBox3.Text = "";
                textBox4.Text = "";
                DGRefresh1();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```



```

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView1.CurrentRow != null)
            {
                Auditorium a = new
Auditorium(Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value),
Convert.ToInt32(dataGridView1.CurrentRow.Cells[1].Value),
Convert.ToInt32(dataGridView1.CurrentRow.Cells[2].Value),
Convert.ToInt32(dataGridView1.CurrentRow.Cells[3].Value));

                db.auditorium.Remove(a);
                db.SaveChanges();
                DGRefresh1();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button3_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView1.CurrentRow != null)
            {
                Auditorium a = new
Auditorium(Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value),
Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text),
Convert.ToInt32(textBox4.Text));

                db.auditorium.Update(a);
                db.SaveChanges();
                DGRefresh1();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

    }

}

private void pictureBox1_Click(object sender, EventArgs e)
{

}

private void button4_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            Groups_of_students a = new
Groups_of_students(Convert.ToInt32(textBox5.Text), textBox6.Text,
textBox7.Text, Convert.ToInt32(textBox8.Text),
Convert.ToInt32(textBox9.Text));

            db.groups_of_students.Add(a);
            db.SaveChanges();
            textBox5.Text = "";
            textBox6.Text = "";
            textBox7.Text = "";
            textBox8.Text = "";
            textBox9.Text = "";
            DGRefresh2();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button5_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView2.CurrentRow != null)
            {
                Groups_of_students a = new
Groups_of_students(Convert.ToInt32(dataGridView2.CurrentRow.Cells[0].Value),
dataGridView2.CurrentRow.Cells[1].Value.ToString(),
dataGridView2.CurrentRow.Cells[2].Value.ToString(),
Convert.ToInt32(dataGridView2.CurrentRow.Cells[3].Value),
Convert.ToInt32(dataGridView2.CurrentRow.Cells[4].Value));

```

```

        db.groups_of_students.Remove(a);
        db.SaveChanges();
        DGRefresh2();
    }
}

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void button6_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView2.CurrentRow != null)
            {
                Groups_of_students a = new
Groups_of_students(Convert.ToInt32(dataGridView2.CurrentRow.Cells[0].Value),
textBox6.Text, textBox7.Text, Convert.ToInt32(textBox8.Text),
Convert.ToInt32(textBox9.Text));

                db.groups_of_students.Update(a);
                db.SaveChanges();
                DGRefresh2();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button7_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            Item a = new Item(Convert.ToInt32(textBox10.Text),
textBox11.Text, Convert.ToInt32(textBox12.Text));

            db.item.Add(a);
            db.SaveChanges();

```

```

        textBox10.Text = "";
        textBox11.Text = "";
        textBox12.Text = "";
        DGRefresh3();
    }

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void button8_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView3.CurrentRow != null)
            {
                Item a = new
Item(Convert.ToInt32(dataGridView3.CurrentRow.Cells[0].Value),
dataGridView3.CurrentRow.Cells[1].Value.ToString(),
Convert.ToInt32(dataGridView3.CurrentRow.Cells[2].Value));

                db.item.Remove(a);
                db.SaveChanges();
                DGRefresh3();
            }
        }

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button9_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView3.CurrentRow != null)
            {
                Item a = new
Item(Convert.ToInt32(dataGridView3.CurrentRow.Cells[0].Value),
textBox11.Text, Convert.ToInt32(textBox12.Text));

```

```

        db.item.Update(a);
        db.SaveChanges();
        DGRefresh3();
    }
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void button10_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            Schedule a = new
Schedule(Convert.ToInt32(textBox13.Text), Convert.ToInt32(textBox14.Text),
Convert.ToInt32(textBox15.Text), Convert.ToInt32(textBox16.Text),
Convert.ToInt32(textBox17.Text), textBox18.Text,
Convert.ToInt32(textBox19.Text), Convert.ToInt32(textBox20.Text));

            db.schedule.Add(a);
            db.SaveChanges();
            textBox13.Text = "";
            textBox14.Text = "";
            textBox15.Text = "";
            textBox16.Text = "";
            textBox17.Text = "";
            textBox18.Text = "";
            textBox19.Text = "";
            textBox20.Text = "";
            DGRefresh4();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button11_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {

```

```

        if (dataGridView4.CurrentRow != null)
        {
            Schedule a = new
Schedule(Convert.ToInt32(dataGridView4.CurrentRow.Cells[0].Value),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[1].Value),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[2].Value),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[3].Value),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[4].Value),
dataGridView4.CurrentRow.Cells[5].Value.ToString(),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[6].Value),
Convert.ToInt32(dataGridView4.CurrentRow.Cells[7].Value));
            db.schedule.Remove(a);
            db.SaveChanges();
            DGRefresh4();
        }
    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button12_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView4.CurrentRow != null)
            {
                Schedule a = new
Schedule(Convert.ToInt32(dataGridView4.CurrentRow.Cells[0].Value),
Convert.ToInt32(textBox14.Text), Convert.ToInt32(textBox15.Text),
Convert.ToInt32(textBox16.Text), Convert.ToInt32(textBox17.Text),
textBox18.Text, Convert.ToInt32(textBox19.Text),
Convert.ToInt32(textBox20.Text));
                db.schedule.Update(a);
                db.SaveChanges();
                DGRefresh4();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

private void button13_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            Staff_of_teachers a = new
Staff_of_teachers(Convert.ToInt32(textBox21.Text), textBox22.Text,
textBox23.Text, textBox24.Text, textBox25.Text, textBox26.Text);

            db.staff_of_teachers.Add(a);
            db.SaveChanges();
            textBox21.Text = "";
            textBox22.Text = "";
            textBox23.Text = "";
            textBox24.Text = "";
            textBox25.Text = "";
            textBox26.Text = "";
            DGRefresh5();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button14_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView5.CurrentRow != null)
            {
                Staff_of_teachers a = new
Staff_of_teachers(Convert.ToInt32(dataGridView5.CurrentRow.Cells[0].Value),
dataGridView5.CurrentRow.Cells[1].Value.ToString(),
dataGridView5.CurrentRow.Cells[2].Value.ToString(),
dataGridView5.CurrentRow.Cells[3].Value.ToString(),
dataGridView5.CurrentRow.Cells[4].Value.ToString(),
dataGridView5.CurrentRow.Cells[5].Value.ToString());
                db.staff_of_teachers.Remove(a);
                db.SaveChanges();
                DGRefresh5();
            }
        }
    }
    catch (Exception ex)

```

```

        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button15_Click(object sender, EventArgs e)
    {
        try
        {
            using (ApplicationContext db = new ApplicationContext())
            {
                if (dataGridView5.CurrentRow != null)
                {
                    Staff_of_teachers a = new
Staff_of_teachers(Convert.ToInt32(dataGridView5.CurrentRow.Cells[0].Value),
textBox22.Text, textBox23.Text, textBox24.Text, textBox25.Text,
textBox26.Text);

                    db.staff_of_teachers.Update(a);
                    db.SaveChanges();
                    DGRefresh5();
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button16_Click(object sender, EventArgs e)
    {
        try
        {
            using (ApplicationContext db = new ApplicationContext())
            {
                Type_of_classes_conducted a = new
Type_of_classes_conducted(Convert.ToInt32(textBox27.Text), textBox28.Text);

                db.type_of_classes_conducted.Add(a);
                db.SaveChanges();
                textBox27.Text = "";
                textBox28.Text = "";
                DGRefresh6();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

```



```

    }

    private void button17_Click(object sender, EventArgs e)
    {
        try
        {
            using (ApplicationContext db = new ApplicationContext())
            {
                if (dataGridView6.CurrentRow != null)
                {
                    Type_of_classes_conducted a = new
Type_of_classes_conducted(Convert.ToInt32(dataGridView6.CurrentRow.Cells[0].V
alue), dataGridView6.CurrentRow.Cells[1].Value.ToString());
                    db.type_of_classes_conducted.Remove(a);
                    db.SaveChanges();
                    DGRefresh6();
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button18_Click(object sender, EventArgs e)
    {
        try
        {
            using (ApplicationContext db = new ApplicationContext())
            {
                if (dataGridView6.CurrentRow != null)
                {
                    Type_of_classes_conducted a = new
Type_of_classes_conducted(Convert.ToInt32(dataGridView6.CurrentRow.Cells[0].V
alue), textBox28.Text);
                    db.type_of_classes_conducted.Update(a);
                    db.SaveChanges();
                    DGRefresh6();
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button19_Click(object sender, EventArgs e)

```

```

{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            Venue a = new Venue(Convert.ToInt32(textBox29.Text),
textBox30.Text, textBox31.Text);

            db.venue.Add(a);
            db.SaveChanges();
            textBox29.Text = "";
            textBox30.Text = "";
            textBox31.Text = "";
            DGRefresh7();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button20_Click(object sender, EventArgs e)
{
    try
    {
        using (ApplicationContext db = new ApplicationContext())
        {
            if (dataGridView7.CurrentRow != null)
            {
                Venue a = new
Venue(Convert.ToInt32(dataGridView7.CurrentRow.Cells[0].Value),
dataGridView7.CurrentRow.Cells[1].Value.ToString(),
dataGridView7.CurrentRow.Cells[2].Value.ToString());
                db.venue.Remove(a);
                db.SaveChanges();
                DGRefresh7();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void button21_Click(object sender, EventArgs e)
{
    try

```

```

        {
            using (ApplicationContext db = new ApplicationContext())
            {
                if (dataGridView7.CurrentRow != null)
                {
                    Venue a = new
Venue(Convert.ToInt32(dataGridView7.CurrentRow.Cells[0].Value),
textBox30.Text, textBox31.Text);
                    db.venue.Update(a);
                    db.SaveChanges();
                    DGRefresh7();
                }
            }

        }
    }
}

```