Министерство образования и науки РФ

Федеральное государственное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Курсовой проект защищен

с оценкой *\_\_\_\_\_\_\_\_*

Преподаватель

\_\_\_\_\_\_\_\_\_\_ Костикова И.Ю.

« » 2020

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «СВЯЗЬ СТАРОСТ ГРУПП СО СЛУЖБАМИ УНИВЕРСИТЕТА»**

Пояснительная записка к курсовому проекту

по дисциплине “Системы управления базами данных”

ЯГТУ 09.03.02-010 КП

Нормоконтролер Проект выполнил

Преподаватель студент группы ЭИС-37

\_\_\_\_\_\_\_\_\_\_ Костикова И.Ю. \_\_\_\_\_\_\_\_\_\_Наугольный Д.И

« » 2020 « » 2020

2020

Министерство образования и науки Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

ЗАДАНИЕ № 10

по курсовому проектированию

Студенту Наугольному Денису Игоревичу

Факультет «Цифровых систем» курс 3 группа ЭИС-37

I. Тема проекта и исходные данные

Разработка информационной системы по теме «СВЯЗЬ СТАРОСТ ГРУПП СО СЛУЖБАМИ УНИВЕРСИТЕТА»

СУБД: MS SQL Server

Среда разработки клиентского приложения: Visual Studio 2019

II. Представить следующие материалы:

1. Текстовые

Пояснительная записка к курсовому проекту, включающая: системный анализ предметной области, концептуальную схему, логическую структуру базы данных, описание приложения, руководство пользователя и техническое задание.

1. Графические

Иллюстрационные материалы представляются студентом на защиту в виде компьютерной презентации в формате MS Power Point

III. Рекомендуемая литература и материал:

1. Ткачев О.А. Создание и манипулирование базами данных средствами СУБД Microsoft SQL Server 2008 [Электронный pecypc]: Учебное пособие/ Ткачев О.А. – Электрон. текстовые данные. – М.: Московский городской педагогический университет, 2013. – 152 с.     – Режим доступа: http://www.bibliocomlectator.ru/book/?id=26613                  .

2. Чубукова И.А. Data Mining [Электронный pecypc]/ Чубукова И.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий  (ИНТУИТ), 2016. – 470 с. – Режим доступа: http://www.bib1iocomp1ectator.ru/book/?id–56315.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IV. Дата выдачи задания \_\_\_\_\_\_\_\_\_

V. Срок сдачи законченного проекта \_\_\_\_\_\_\_\_\_

VI. Отметка о явке на консультацию:

1)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 5)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 6)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель проекта\_\_\_\_\_\_ Михайлов Д. О. и. о. зав. кафедрой\_\_\_\_\_\_С. Ю. Бойков

Задание принял к исполнению «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись студента)

Реферат

40 с., 31 рис., 15 табл., 5 источника, 3 прил.

MS SQL SERVER, СУБД, ИНФОРМАЦИОННАЯ СИСТЕМА, КЛИЕНТ-СЕРВЕРНАЯ АРХИТЕКТУРА, С#.

Объектом исследования является комплексная информационная система для контроля и организации расписания.

Было необходимо разработать информационную подсистему СВЯЗЬ СТАРОСТ ГРУПП СО СЛУЖБАМИ УНИВЕРСИТЕТА. Университету необходимо автоматизировать составление расписания. Подсистема должна иметь возможность печатать отчеты, формировать расписание, вести учет посещаемости и т.д. так же подсистема должна обеспечивать формирование следующих печатных форм: «Отчет о посещаемости».

Оглавление

[Введение 5](#_Toc61711753)

[1. Аналитическая часть 7](#_Toc61711754)

[1.1 Системный анализ предметной области 7](#_Toc61711755)

[1.2 Назначение проектируемой системы 8](#_Toc61711756)

[1.3 Цели создания проектируемой системы 9](#_Toc61711757)

[1.4 Задачи, решаемые с помощью проектируемой системы 9](#_Toc61711758)

[2. Проектная часть 10](#_Toc61711759)

[2.1 Функциональная модель 10](#_Toc61711760)

[2.2 UML – проектирование 12](#_Toc61711761)

[2.2.1 Диаграмма вариантов использования 12](#_Toc61711762)

[2.2.2 Диаграмма классов 13](#_Toc61711763)

[2.2.3 Диаграмма развертывания 13](#_Toc61711764)

[2.3 Проектирование базы данных 14](#_Toc61711765)

[2.3.1 Концептуальное проектирование 14](#_Toc61711766)

[2.3.2 Логическое проектирование 14](#_Toc61711767)

[2.3.3 Физическое проектирование 15](#_Toc61711768)

[2.3.4 Роли и права доступа к данным 18](#_Toc61711769)

[2.4 Проектирование интерфейса клиентской части 18](#_Toc61711770)

[3. Технологическая часть 19](#_Toc61711771)

[3.1 Выбор программных средств 19](#_Toc61711772)

[3.1.1 Выбор СУБД 19](#_Toc61711773)

[3.1.2 Выбор средств разработки 19](#_Toc61711774)

[3.2 Описание работы с программой 19](#_Toc61711775)

[3.3 Описание тест – кейсов 28](#_Toc61711776)

[Заключение 30](#_Toc61711777)

[Список используемых источников и литературы 31](#_Toc61711778)

[Приложение А. 32](#_Toc61711779)

[Приложение Б. 37](#_Toc61711780)

[Приложение В 39](#_Toc61711781)

# Введение

В последние годы бурное развитие новых технологий сделали актуальным использование баз данных в организациях различных форм собственности и направлений деятельности. Успехи в исследованиях и разработке баз данных стали основой фундаментальных разработок коммуникационных систем, транспорта и логистики, финансового менеджмента, методов доступа к научной литературе, а также многих гражданских и военных приложений.  
  
Базы данных лежат в основе информационных систем, и это коренным образом изменило характер работы многих предприятий и организаций. Цель базы данных – помочь людям и организациям вести учет, осуществлять информационную поддержку при принятии решений. Базы данных представляют собой особую организацию данных и предполагают использование специальных программно-языковых средств. Базы данных - это сложные взаимосвязанные информационные образования, предназначенные для многоцелевого использования.  
  
Проектирование и разработка базы данных на основе реляционных моделей данных требуют, с одной стороны, знания предметной области, а с другой, владения современными информационными технологиями. Структурная независимость реляционной базы данных и ее независимость по данным позволяет исследовать логическую структуру модели без обращения к физическим аспектам хранения и извлечения данных. Одна из самых важных причин простоты реляционной модели базы данных состоит в том, что она отвечает на вопрос какие данные необходимо извлечь, а не как извлечь данные.  
  
Целью выполнения курсового проекта является систематизация, закрепление и углубление теоретических знаний и практических навыков проектирования баз данных и управления ими.  
  
Курсовой проект включает в себя: постановку задачи, обоснование необходимости создания БД, описание этапов нормализации БД, обоснование выбор используемых программных средств.  
  
Важнейшим элементом в процессе разработки приложения является выбор правильной IDE, зависящей не только от платформы, но и от уровня собственной подготовки. Интегрированная среда разработки — это некий комплекс отдельных инструментов разработчика, позволяющий не только написать код программы, но и скомпилировать этот код в готовое приложение, провести тестирование этой программы, поиск ошибок, рефакторинг и другие действия. Сейчас существует множество профессиональных продуктов для разработки приложений.

Visual Studio – это среда разработки от Microsoft. Visual Studio в основном известна для написания приложений, включающих в себя .NET. Это полный набор инструментов, позволяющий произвести точную отладку и настройку приложения.

Visual Studio предназначена не только для разработчиков на C++, но также поддерживает многие другие популярные языки, такие как C#, Visual Basic и F#.

Visual Studio предлагает множество функций, некоторые из которых: интеллектуальное автодополнение кода; дизайнер графических форм (GUI); простая в использовании навигационная система.

Вы можете использовать IDE для разработки компьютерных программ для Microsoft Windows, а также веб-сайтов, веб-приложений и веб-сервисов.

Данная среда является интегрированной, так как в ней возможна совместная работа над одним проектом с разных компьютеров в режиме реального времени.

**Главными преимуществами Microsoft Visual Studio являются:** большое количество функций; наличие отладчика для JS; поддержка нескольких мониторов; использование сплит-окон XML и CSS; удобная регулировка рабочей области; продвинутый конструктор и редактор кода; простой и интуитивно понятный интерфейс на русском языке; возможность разрабатывать графические элементы Windows Forms; регулярные автообновления от официального сайта Майкрософт.

Исходя из перечисленного можно сделать вывод, что Visual Studio – среда, дающая широкие возможности в разработке программного обеспечения, что делает её идеальным выбором для реализации проекта.

# Аналитическая часть

# Системный анализ предметной области

В системе необходимо наличие ролей следующих типов:

- «Староста»

- «Деканат»

- «Преподаватель»

Необходимо реализовать следующий функционал для пользователя с ролью «Староста»:

- просмотр расписания группы.

- контроль посещаемости занятий и отправка отчётов пользователю с ролью «Преподаватель»

Необходимо реализовать следующий функционал для пользователя с ролью «Преподаватель»:

- изменение расписания (с автоматическим уведомлением пользователей типа «Староста», которых затрагивает это изменение);

- контроль посещаемости занятий

- назначение дат зачётов и переэкзаменовок.

- просмотр личного расписания.

- просмотр занятости аудиторий

Необходимо реализовать следующий функционал для пользователя с ролью «Деканат»:

- просмотр и редактирование расписания с уведомлениями преподавателей и студентов;

- формирование списков студентов.

Необходимо также реализовать следующую логику: если меняется расписание, то после авторизации преподавателя или студента, расписание которого было изменено, ему должно отображаться оповещение об изменениях (например, в виде всплывающего окна).

Обеспечить формирование следующих печатных форм: «Отчет о посещаемости».

Базовые сущности в этой предметной области:

Сущность «Пользователь»:

- id пользователя

- id\_роли

- ФИО

- Логин

- Пароль

- Email

В БД должна храниться информация о кафедрах, студенческих группах и студентах.

Сущность «Кафедра»:

- id кафедры

- название кафедры

Сущность «Группа»:

- id группы

- Название группы

- Курс

- id кафедры

Сущность «Студент»:

- id\_студента

- id\_пользователя

- id\_группы

- этоСтароста (столбец с логическим типом данных, если true – значит, что студент является старостой своей группы)

Сущность «Преподаватель»:

- id\_преподавателя

- id\_пользователя

- id\_кафедры

Сущность «Сотрудник Деканата»:

- id\_сотрудника

- id\_пользователя

Нужно хранить информацию о расписании. Поэтому введём сущность «Дисциплина».

Сущность «Дисциплина»:

- id дисциплины

- название

В расписании указывается тип занятия. Поэтому введём сущность «Тип занятия» (лекция/практика/лабораторная работа). Также введём сущность «Аудитории».

Сущность «Тип занятия»:

- id типа занятия

- название типа

Сущность «Аудитория»:

- id\_аудитории

- имя

А теперь введём сущность «Расписание»:

- id\_занятия

- id\_дисциплины

- id\_преподавателя

- id\_типа занятия

- id\_группы

- id\_аудитории

- ДатаВремя начала занятия

- ДатаВремя конца занятия

- id\_сотрудника\_деканата

- ДатаВремя изменения

Т.к. сотрудники деканата добавляют и меняют расписание, то вводится столбец «айди сотрудника деканата» - тот сотрудник, который внёс измененияю. И датавремя последнего изменения расписания.

Нам надо хранить информацию о посещаемости студентами пар.

Сущность «Посещаемость»:

- id\_посещаемости

- id\_занятия

- id\_студента

- присутствие (столбец с логическим типом)

Нужно хранить информацию о расписании зачётов, экзаменов и так далее.

Сущность «РасписаниеКонтроля».

Но перед этим выделим сущность «Тип контроля» (зачёт, дифференцированный зачёт, экзамен, пересдача зачёта, пересдача экзамена)

Сущность «Тип контроля»:

- id\_типа\_контроля

- название типа

# Назначение проектируемой системы

Общее описание:

Как объект взят университет, в котором осуществляется связь старост групп и служб университета.

БД должна содержать:

1. Данные о расписании.

2. Данные о посещаемости занятий.

3. Данные о занятости аудиторий.

4. Данные о датах зачетом и переэкзаменовок.

5. Данные о списках студентов

Бизнес-правила:

* У старосты может отсутствовать отчество.
* У студента может отсутствовать отчество.
* У преподавателя может отсутствовать отчество.
* Контактные номера студентов не должны совпадать.
* Контактные номера старост не должны совпадать.
* Контактные номера преподавателя не должны совпадать.

# Цели создания проектируемой системы

Информационная система «СВЯЗЬ СТАРОСТ ГРУПП СО СЛУЖБАМИ УНИВЕРСИТЕТА» должна обеспечивать:

1. Формирование расписания.
2. Учет посещаемости.
3. Регистрация новых пользователей, изменение их данных.
4. Уведомление пользователей о изменении расписания.
5. Возможность просмотра информации о занятости аудиторий.
6. Формирование отчетности о посещаемости.
7. Формирование списков студентов.

Цель разработки системы – автоматизация связи старост групп со службами университета.

# Задачи, решаемые с помощью проектируемой системы

С помощью данной проектируемой системы работа производства будет осуществляться проще и эффективней. Люди, работающие с системой, смогут вносить изменения в данные справочников в базе данных. Благодаря проектируемой системе можно будет организовывать расписание.

# Проектная часть

# Функциональная модель

Деятельность системы показана на рисунке 1. В систему заносятся данные о преподавателях, аудиториях, дисциплинах и студентах. Сотрудники формируют расписание в зависимости о занятости аудиторий и списков студентов.

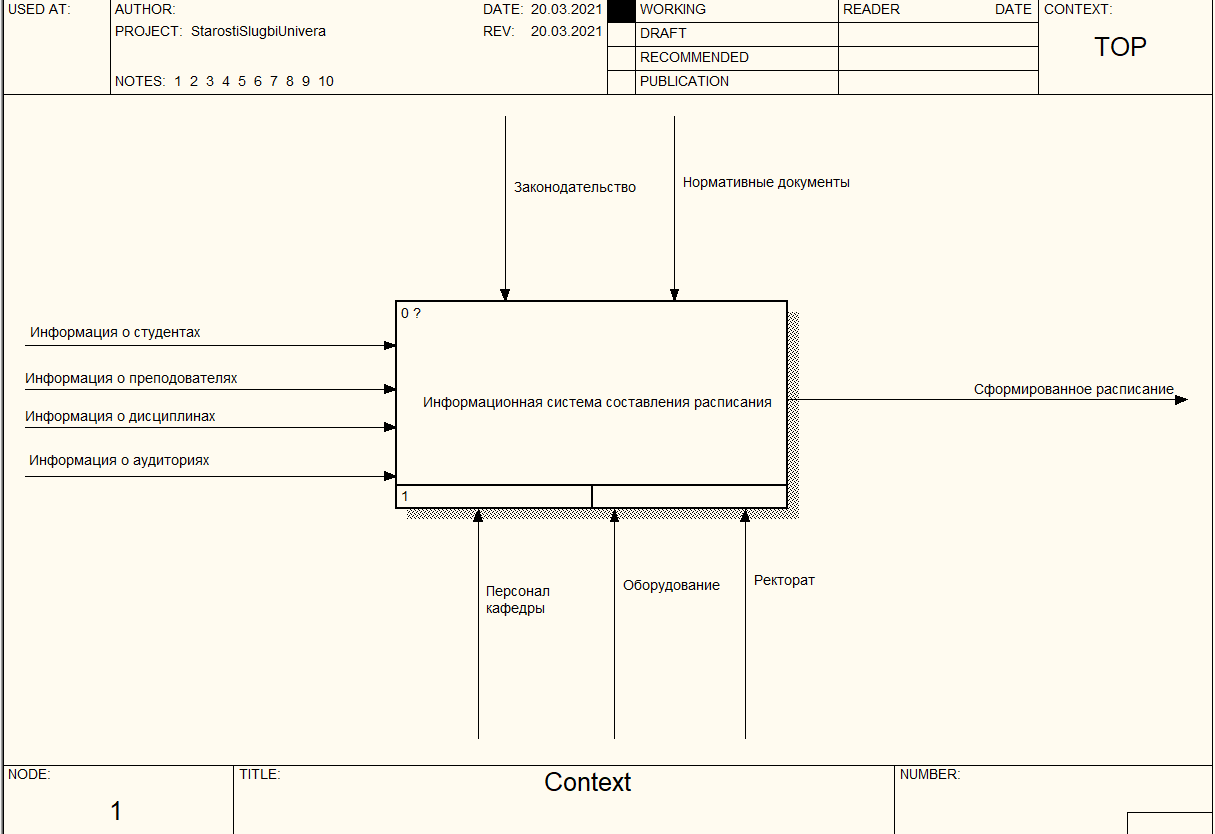


Рисунок 1 – Контекстная диаграмма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название блока | Входы | Выходы | Управление | Механизм |
| Информационная система составления расписания | Информация о студентах | Сформированное расписание | Законодательство | Персонал  кафедры |
| Информация о преподавателях |  | Нормативные документы | Ректорат |
| Информация о дисциплинах |  |  | оборудование |
| Информация о аудиториях |  |  |  |

Таблица 1 – Описание контекстной диаграммы

На рисунке 2 более подробно представлен процесс деятельности системы. Сначала формируется план учебной работы, затем распределяется учебная нагрузка, принимаются индивидуальные корректировки расписания, и в конце происходит формирование конечного утвержденного расписания.

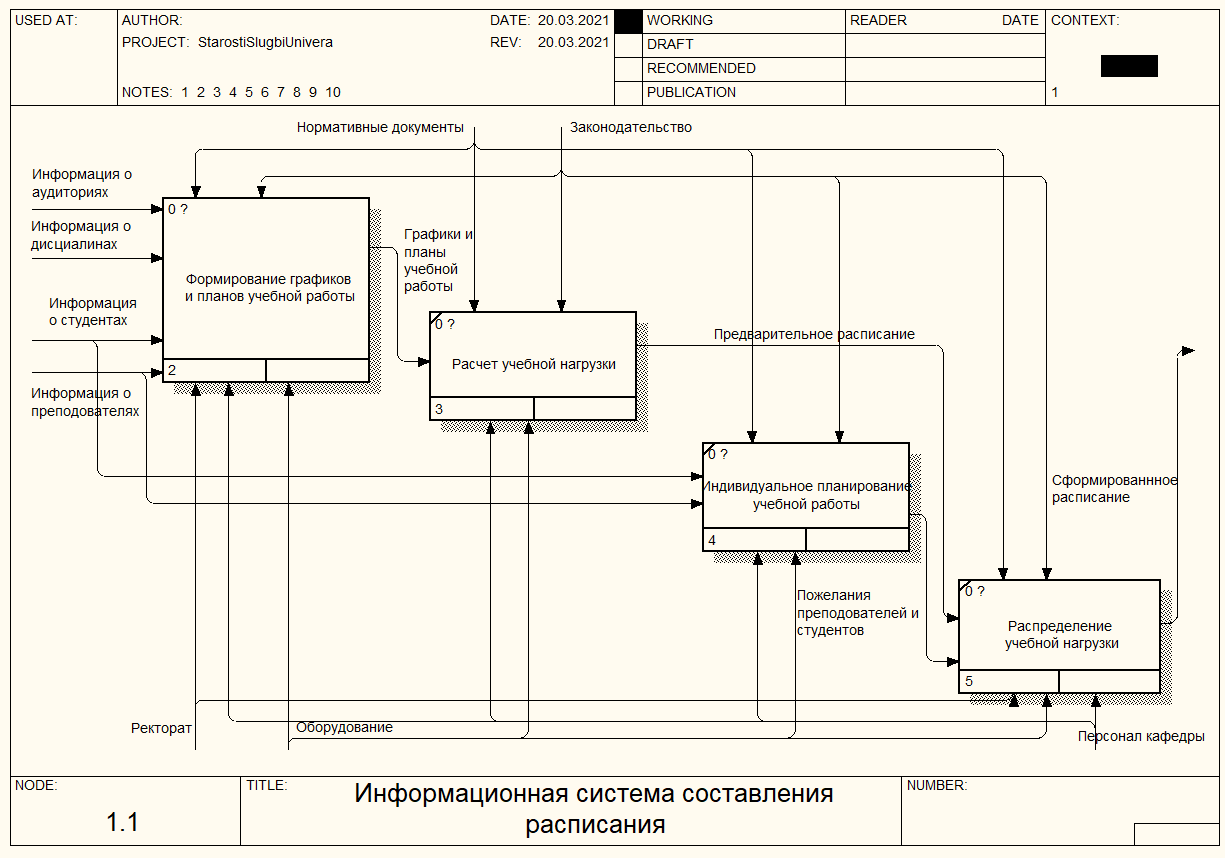


Рисунок 2 – Диаграмма декомпозиции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название блока | Входы | Выходы | Управление | Механизм |
| Формирование  поставки | Информация о студентах | Сформированное расписание | Законодательство | Персонал |
| Информация о преподавателях |  | Нормативные документы | Ректорат |
| Информация о дисциплинах |  |  | оборудование |
| Информация о аудиториях |  |  |  |

Таблица 2 – Описание диаграммы декомпозиции

На рисунке 3 более подробно представлен процесс формирования графиков и планов учебной работы. Сначала анализируются стандарты дисциплин, затем по равномерно распределяют учебные часы студентов и преподавателей, и в конце формируется предварительный план учебной работы.

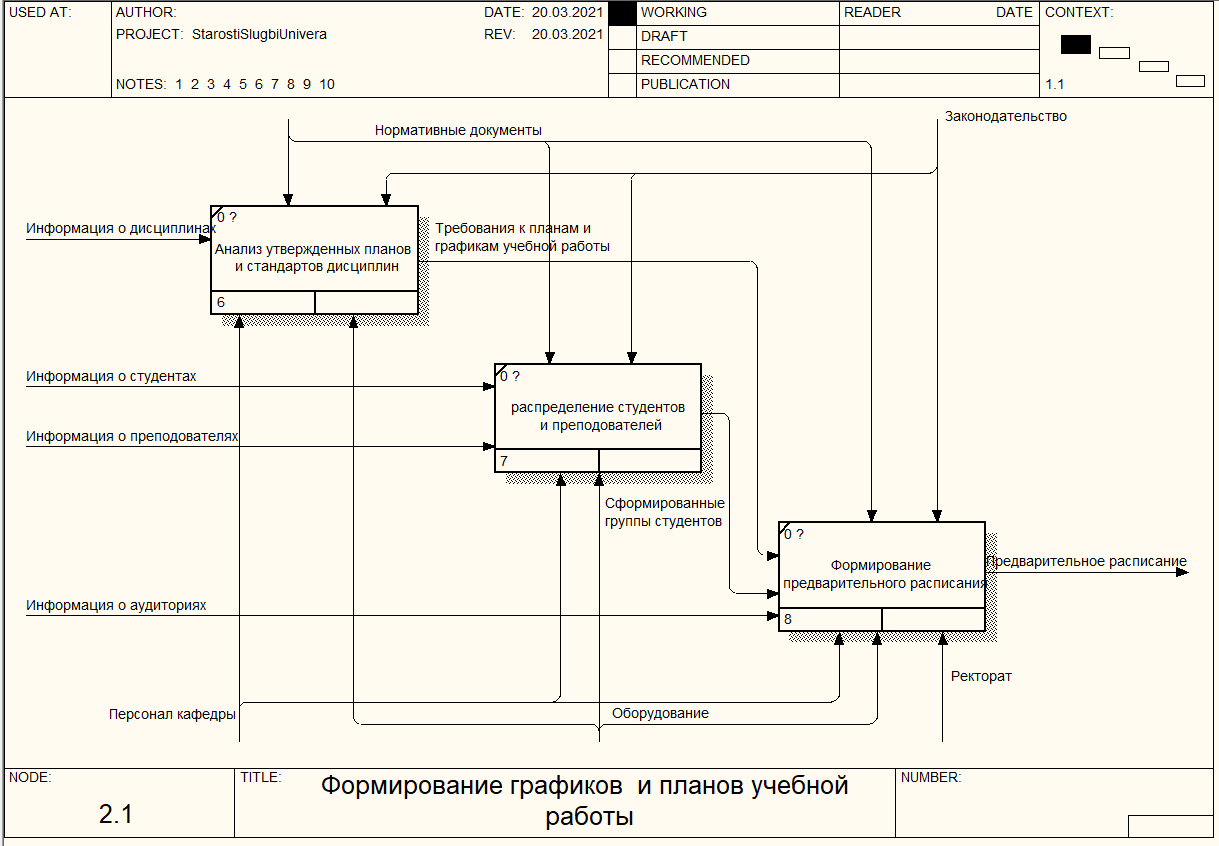


Рисунок 3 – Диаграмма декомпозиции 2

# UML – проектирование

# Диаграмма вариантов использования

Для любых действий в системе СВЯЗЬ СТАРОСТ ГРУПП СО СЛУЖБАМИ УНИВЕРСИТЕТА пользователь сначала должен пройти авторизацию и действовать согласно правам, которые он имеет. Возможности для Администратора, Сотрудника Деканата, Преподавателя, Старосты и Студента описаны на рисунках 3 – 8.

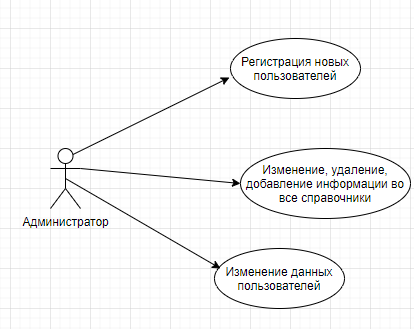


Рисунок 4 – Диаграмма вариантов использования для Администратора

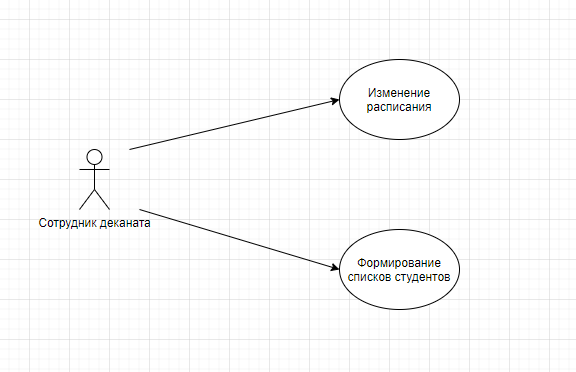


Рисунок 5 – Диаграмма вариантов использования для Сотрудника деканата

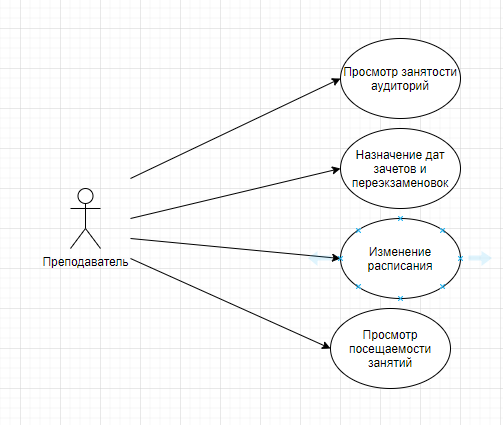


Рисунок 6 – Диаграмма вариантов использования для Преподавателя

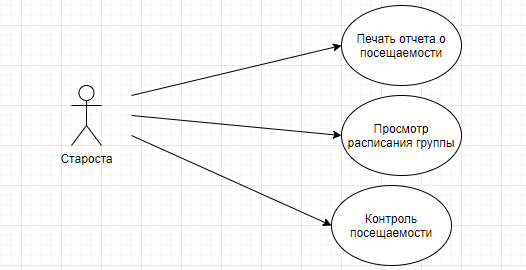


Рисунок 7 – Диаграмма вариантов использования для Старосты

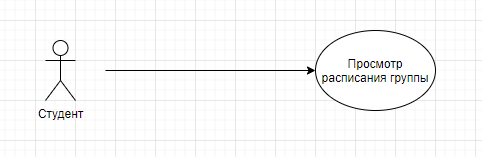


Рисунок 8 – Диаграмма вариантов использования для Студента

# Диаграмма классов

Диаграмма классов является основным логическим представлением модели и содержит детальную информацию о внутреннем устройстве данной БД. Все это отображено на рисунке 8.

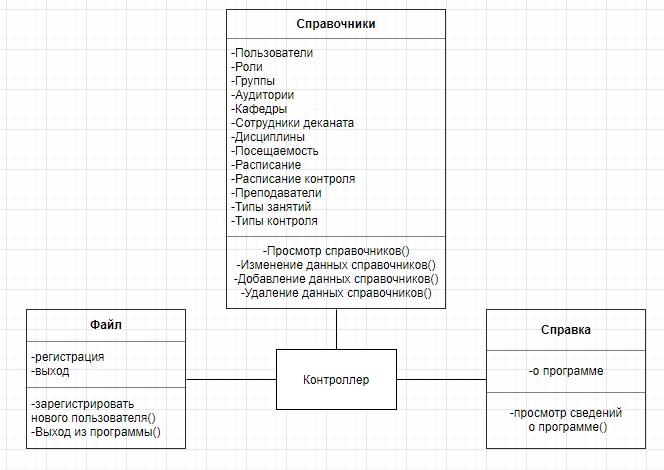


Рисунок 9 – Диаграмма классов

# Диаграмма развертывания

Диаграмма развертывания предназначена для того, чтобы видеть, как будет распределена система. Это показано на рисунке 9.

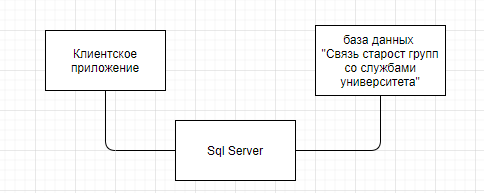


Рисунок 10 – Диаграмма развертывания

# Проектирование базы данных

# Концептуальное проектирование

Концептуальное проектирования – это создание схемы БД, включающей определение важнейших сущностей и связей между ними. Сущности в данной БД: «Роль», «Пользователь», «Расписание», «Расписание Контроля», «Аудитория», «Группа», «Тип занятия», «Детали учета» и т.д.. Каждая сущность содержит атрибуты – набор данных, который характеризует сущность. Так же все сущности имеют связи – отношения, графически отображаемые ассоциации в виде стрелки. Концептуальная модель для данной БД показана на рисунке 11.

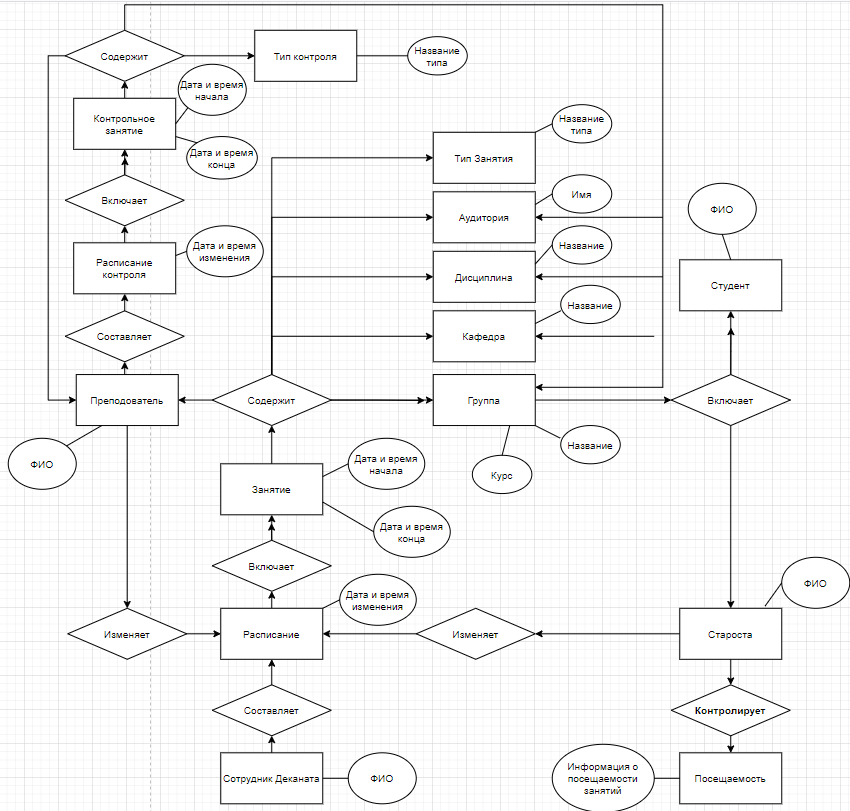


Рисунок 11 – Концептуальная модель

# Логическое проектирование

Целью логического проектирования является развитие концептуальной модели и преобразование в вид выбранной модели данных. Для начала определяем таблицы и их документирование. Количество сущностей, выделенных в концептуальной модели равно количеству таблиц в логической модели. А количество атрибутов в каждой сущности равно количеству столбцов в этих таблицах. Таким образом сначала создается первая нормальная форма, представленная на рисунке 12, в такой форме все столбцы должны быть атомарными.

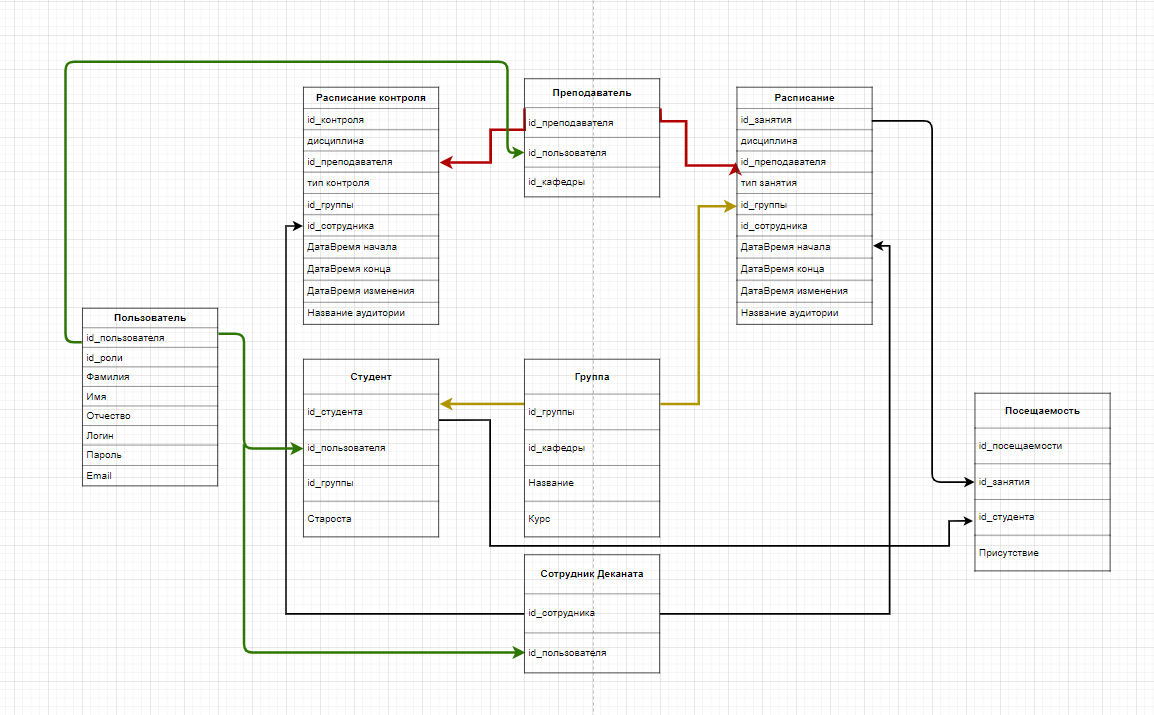


Рисунок 12 – 1 Нормальная форма

Далее составляется вторая и третья нормальные формы. В таких формах каждый столбец, не входящий в состав первичного ключа, должен зависеть от всего первичного ключа, а не от отдельных частей. 3 нормальная форма для данной БД представлена на рисунке 13.

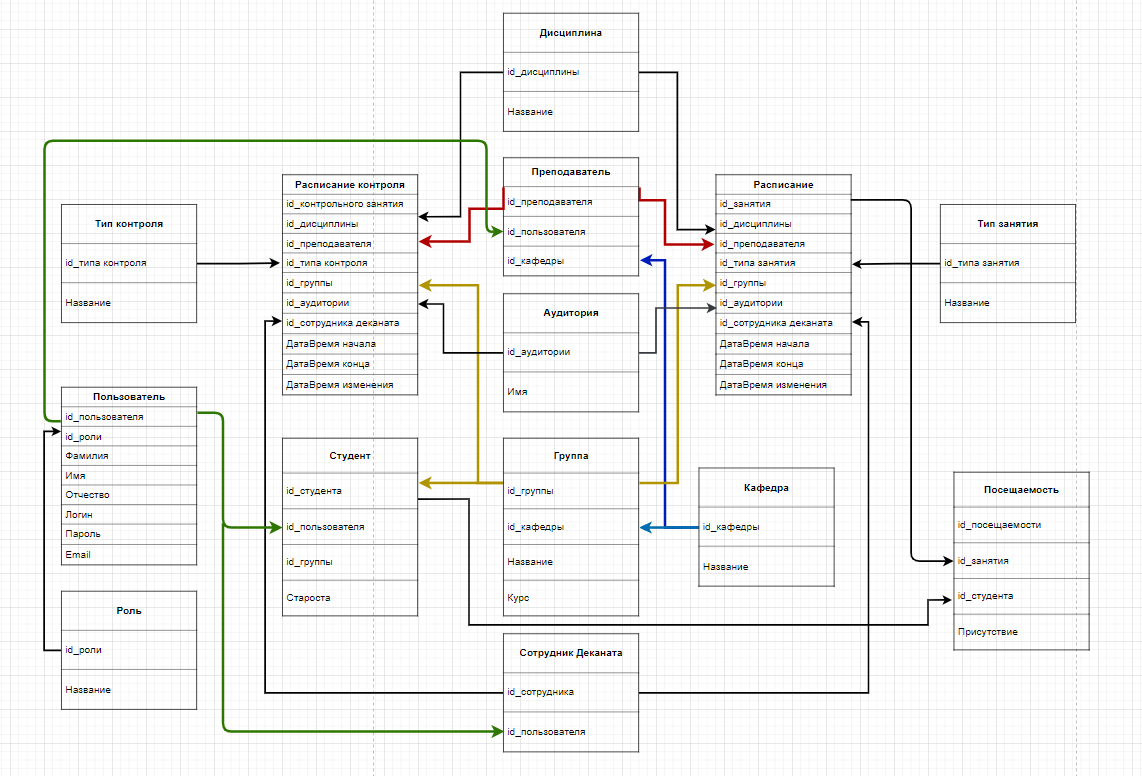


Рисунок 13 – 3 Нормальная форма

# Физическое проектирование

Физическое проектирование – определение особенностей хранения данных и методов доступа к ним. На данном этапе происходит назначение каждому столбцу каждой таблицы определенного типа данных, установка значений по умолчанию, добавление ограничений и определение правил автоматического увеличения некоторых столбцов. Ниже в таблицах представлено описание атрибутов сущностей для данной БД.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Роль | Id\_роли | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор роли |
| Название | Да | VARCHAR(10) | - | Имя роли |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Пользователи | Id\_пользователя | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор пользователя |
| Id\_роли | Да | INT | Внешний ключ для связи с таблицей “Роль” | Идентификатор роли |
| Фамилия | Да | VARCHAR(30) | - | Фамилия с пользователя |
| Имя | Да | VARCHAR(30) | - | Имя пользователя |
| Отчество | Нет | VARCHAR(30) | - | Отчество пользователя |
| Номер  мобильного | Да | VARCHAR(11) | Формат: 79991112233  Без других символов. | Контактный мобильный телефон пользователя. |
| Логин | Да | VARCHAR(255) | Строка в нижнем регистре, состоящая из символов латиницы и цифр. | Логин пользователя для авторизации в приложении |
| Пароль | Да | VARCHAR(255) | Строка, состоящая из символов латиницы и цифр. | Пароль учётной записи пользователя для авторизации в приложении |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Студент | Id\_студента | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор студента |
| Id\_пользователя | Да | INT | Внешний ключ для связи с таблицей “Пользователь” | Идентификатор пользователя |
| Id\_группы | Да | INT | Внешний ключ для связи с таблицей “группа” | Идентификатор группы |
| Староста | Да | BIT | - | Является ли старостой |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Группа | Id\_ группы | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор группы |
| Id\_кафедры | Да | INT | Внешний ключ для связи с таблицей “кафедра” | Идентификатор кафедры |
| Название | Да | VARCHAR(30) | - | Название группы |
| курс | Да | TINYINT(5) | - | Номер курса |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Кафедра | Id\_кафедры | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор кафедры |
| Название | Да | VARCHAR(10) | - | Имя кафедры |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Расписание | Id\_занятия | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор расписания |
| Id\_дисциплины | Да | INT | Внешний ключ для связи с таблицей “дисциплина” | Идентификатор дисциплины |
| Id\_преподавателя | Да | INT | Внешний ключ для связи с таблицей “преподаватель" | Идентификатор преподавателя |
| Id\_типа занятия | Да | INT | Внешний ключ для связи с таблицей “Тип занятия | Идентификатор типа занятия |
| Id\_группы | Да | INT | Внешний ключ для связи с таблицей “группа ” | Идентификатор группы |
| Id\_аудитории | Да | INT | Внешний ключ для связи с таблицей “аудитория ” | Идентификатор аудитории |
| Id\_сотрудника деканата | Да | INT | Внешний ключ для связи с таблицей “сотрудник деканата ” | Идентификатор сотрудника деканата |
| ДатаВремя начала | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |
| ДатаВремя конца | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |
| ДатаВремя изменения | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Тип занятия | Id\_ Типа занятия | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор Типа занятия |
| Название | Да | VARCHAR(10) | - | Имя типа занятия |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Аудитория | Id\_ Аудитории | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор Аудитории |
| Имя | Да | VARCHAR(10) | - | Имя аудитории |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Преподаватель | Id\_ Преподавателя | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор Преподавателя |
| Id\_пользователя | Да | INT | Внешний ключ для связи с таблицей “Пользователь” | Идентификатор пользователя |
| Id\_кафедры | Да | INT | Внешний ключ для связи с таблицей “кафедра” | Идентификатор кафедры |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Дисциплина | Id\_ Дисциплины | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор Дисциплины |
| Название | Да | VARCHAR(10) | - | Название дисциплины |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Посещаемость | Id\_ посещаемости | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор посещаемости |
| Id\_занятия | Да | INT | Внешний ключ для связи с таблицей “Расписание” | Идентификатор занятия |
| Id\_студента | Да | INT | Внешний ключ для связи с таблицей “Студент” | Идентификатор студента |
| присутствие | Да | BIT | - | Присутствует ли на занятии |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Сотрудник деканата | Id\_ Сотрудника деканата | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор Сотрудника деканата |
| Id\_пользователя | Да | INT | Внешний ключ для связи с таблицей “Пользователь” | Идентификатор пользователя |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Расписание контроля | Id\_контрольного\_занятия | Да | INT с атрибутом IDENTITY | Первичный ключ. Автоматическое увеличение на 1. | Идентификатор расписания |
| Id\_дисциплины | Да | INT | Внешний ключ для связи с таблицей “дисциплина” | Идентификатор дисциплины |
| Id\_преподавателя | Да | INT | Внешний ключ для связи с таблицей “преподаватель" | Идентификатор преподавателя |
| Id\_типа контроля | Да | INT | Внешний ключ для связи с таблицей “Тип контроля” | Идентификатор типа контроля |
| Id\_группы | Да | INT | Внешний ключ для связи с таблицей “группа ” | Идентификатор группы |
| Id\_аудитории | Да | INT | Внешний ключ для связи с таблицей “аудитория ” | Идентификатор аудитории |
| Id\_сотрудника деканата | Да | INT | Внешний ключ для связи с таблицей “сотрудник деканата ” | Идентификатор сотрудника деканата |
| ДатаВремя начала | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |
| ДатаВремя конца | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |
| ДатаВремя изменения | Да | DATETIME | Формат: DD.MM.YYYY HH:MM:SS | Дата и время |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица | Столбец | Обязательность | Тип данных | Ограничения | Описание |
| Тип контроля | Id\_ Типа контроля | Да | INT с атрибутом IDENTITY | Первичный ключ.  Автоматическое увеличение на 1. | Идентификатор Типа контроля |
| Название | Да | VARCHAR(10) | - | Имя типа контроля |

# Роли и права доступа к данным

На сервере были созданы 5 пользователя. Первый – Администратор с ролью db\_owner, второй – Сотрудник деканата с ролью DecanatRole, третий– Преподаватель с ролью TeacherRole, четвертый – Староста с ролью StarostaRole, пятый – Студент с ролью StudentRole . В системе используется проверка подлинности Windows.

# Проектирование интерфейса клиентской части

Для начала пользователю предлагается пройти авторизацию для входа в систему, как показано на рисунке 14:

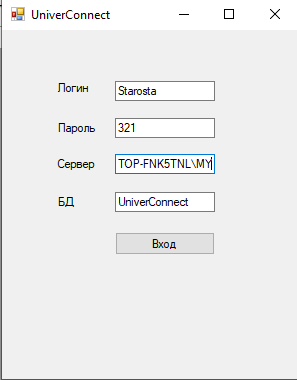


Рисунок 14 – Вход в систему

После входа в зависимости от пользователя откроется одна из 5 основных MDI форм. Все это видно на рисунке 14-19:

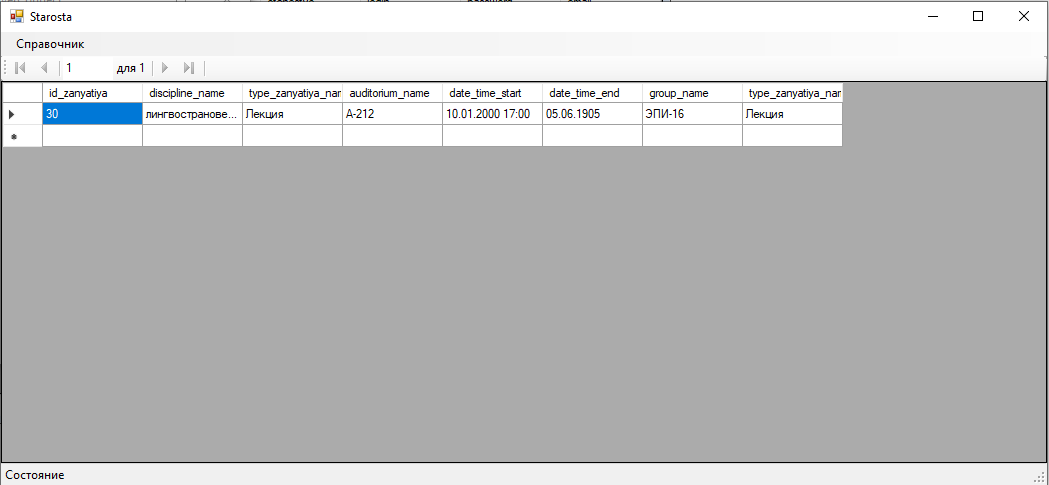


Рисунок 15 – Вход в систему пользователя Starosta

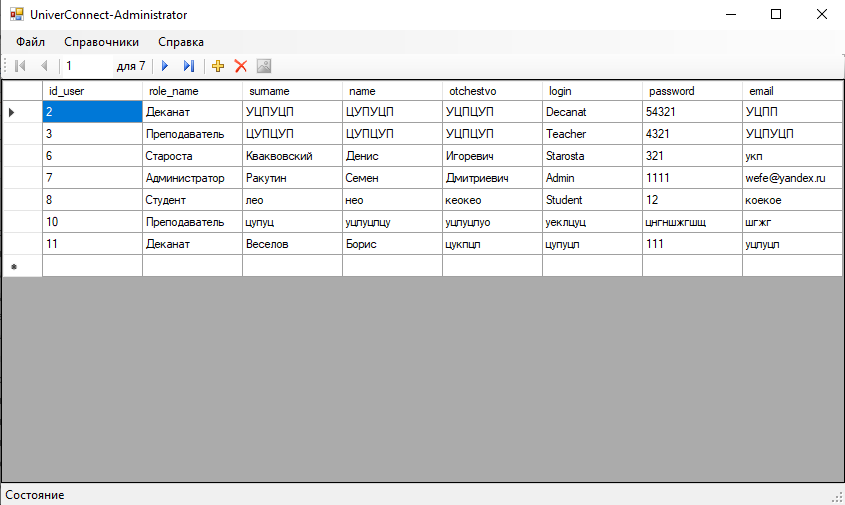


Рисунок 16 – Вход в систему пользователя Admin

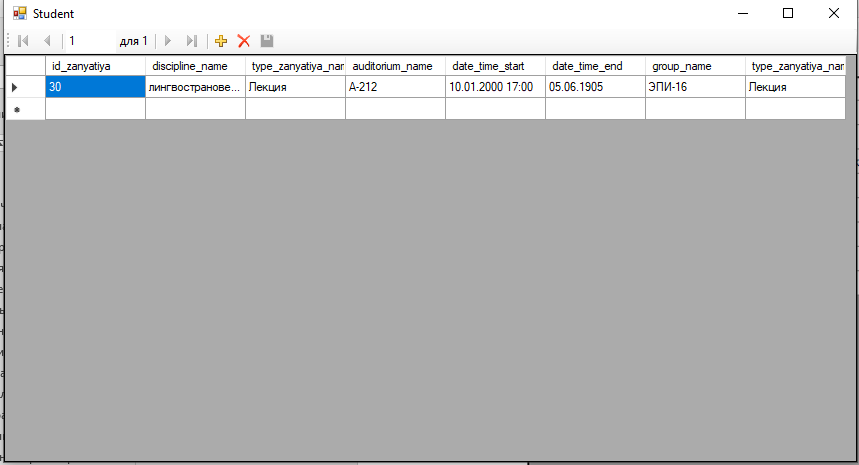


Рисунок 17 – Вход в систему пользователя Student

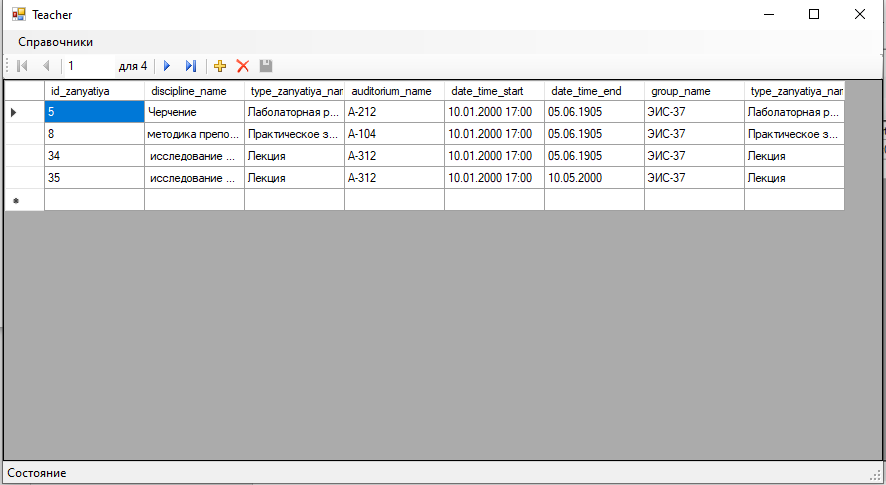


Рисунок 18 – Вход в систему пользователя Teacher

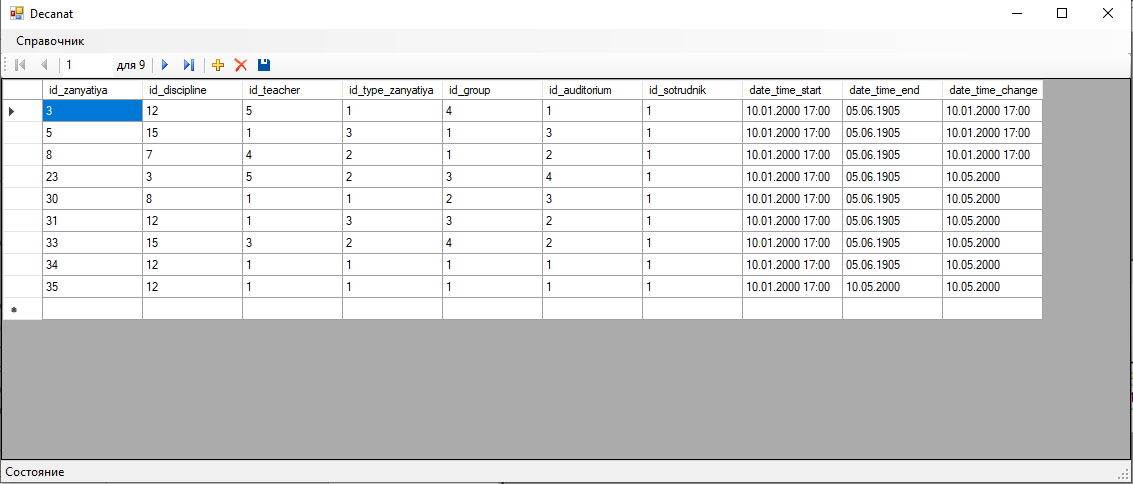


Рисунок 19 – Вход в систему пользователя Decanat

# Технологическая часть

# Выбор программных средств

# Выбор СУБД

Для выполнения курсового проекта мною был выбран СУБД Microsoft SQL Server Management Studio. Я сделала такой выбор по следующим причинам:

1. Данная версия программы распространяется бесплатно
2. Данная программа предоставляет все необходимые средства для создания курсового проекта
3. Данная СУБД была рекомендованной в рамках создания этого курсового проекта.

# Выбор средств разработки

В качестве среды разработки курсового проекта мною была выбрана Microsoft Visual Studio 2019 и язык С#. На результат моего выбора повлияли следующие факторы:

1. Язык программирования С# достаточно хорошо мне известен и понятен
2. Microsoft Visual Studio позволяет пользователям создавать приложения с удобным интерфейсом
3. Понятный и удобный интерфейс Microsoft Visual Studio.

# Описание работы с программой

После того, как пользователь авторизовался как администратор он может начать работать с функционалом приложения. При нажатии на кнопку Файл из выпадающего списка можно выбрать Регистрация как на рисунке13:

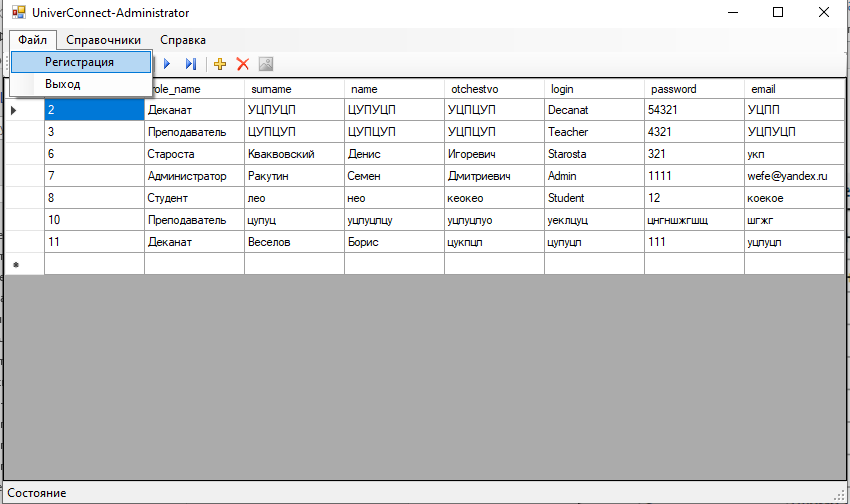


Рисунок 20 – Меню Файл

Регистрируем нового пользователя, как показано на рисунке 14 и вписываем имя пользователя, логин, пароль и его роль в БД. Если нажать на кнопку Регистрация пользователь будет зарегистрирован:

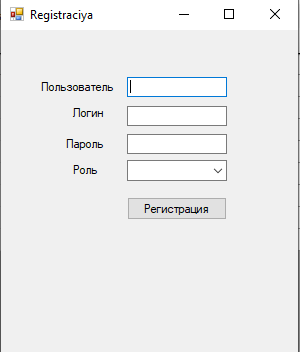


Рисунок 21 – Регистрация нового пользователя

Нажав на кнопку Справочники выводится меню справочников как на рисунке 21. В любом выбранном справочнике можно внести изменения в данные. Можно удалить, добавить или изменить данные и сохранить эти изменения в БД. Вид справочников представлен на рисунках 21 – 34:

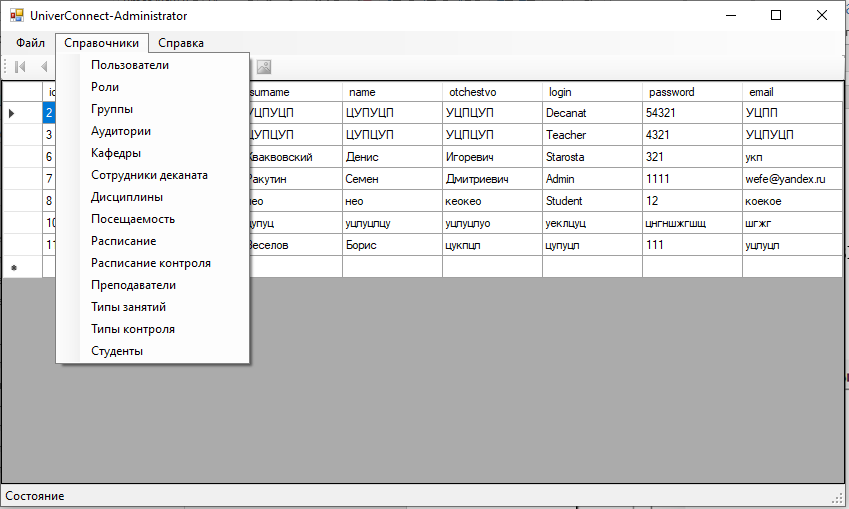


Рисунок 22 – Выбор справочника

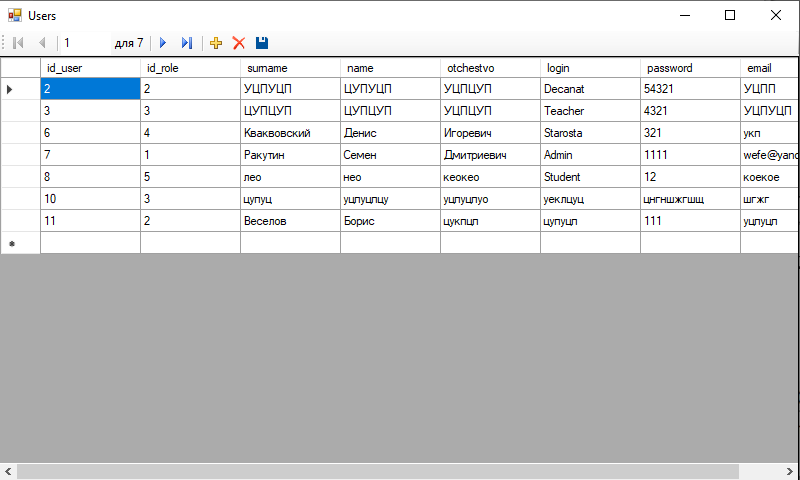


Рисунок 23 – Справочник пользователи

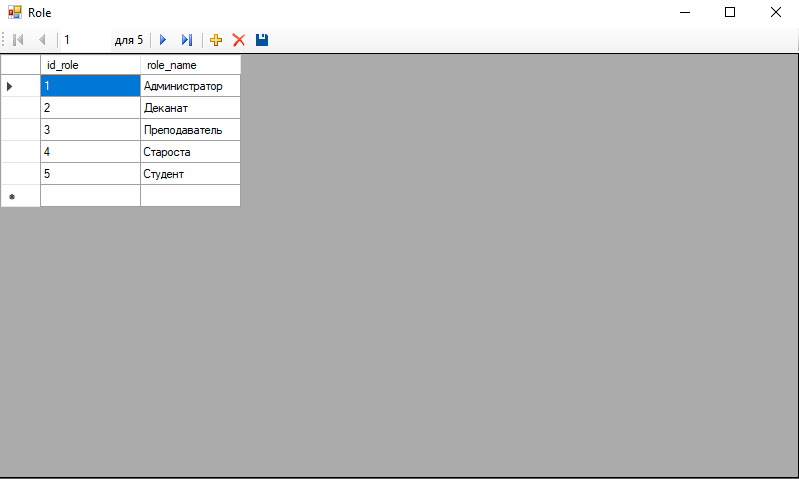


Рисунок 24 – Справочник роли

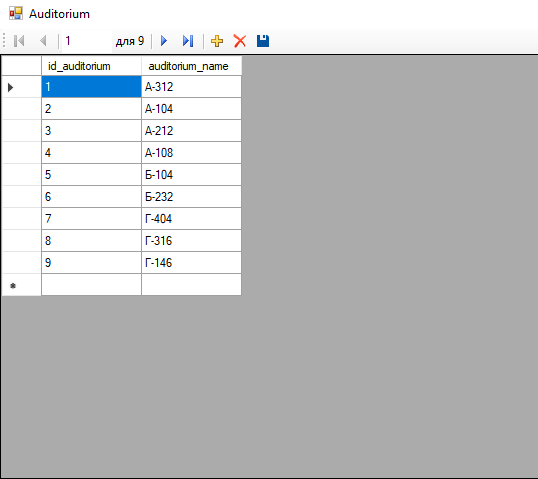


Рисунок 25 – Справочник аудитории

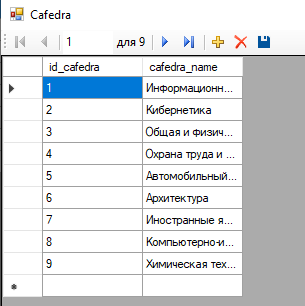


Рисунок 26 – Справочник кафедры

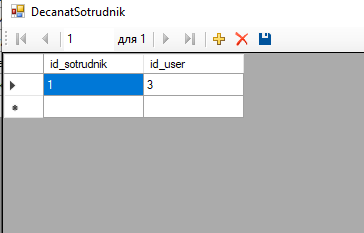


Рисунок 27 – Справочник сотрудники деканата

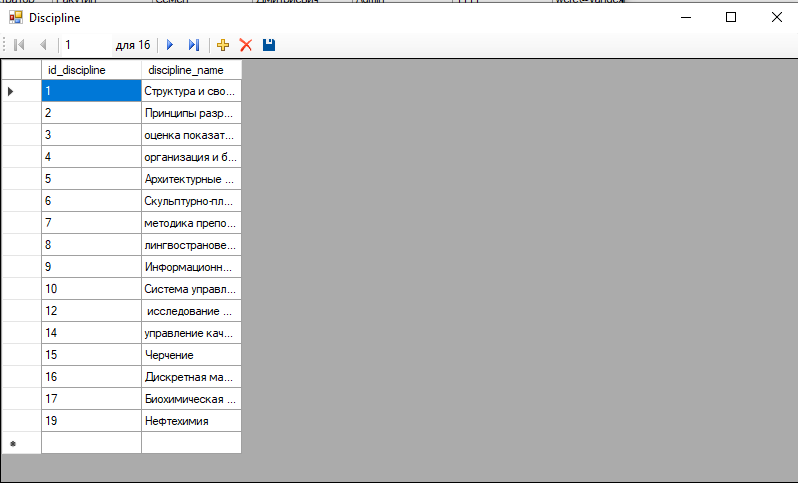


Рисунок 28 – Справочник дисциплины

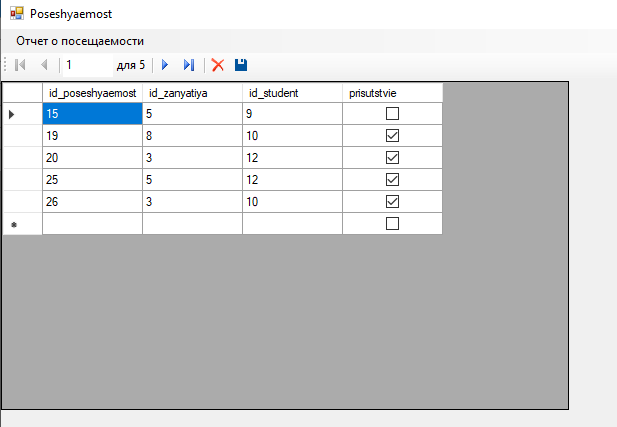


Рисунок 29 – Справочник посещаемость

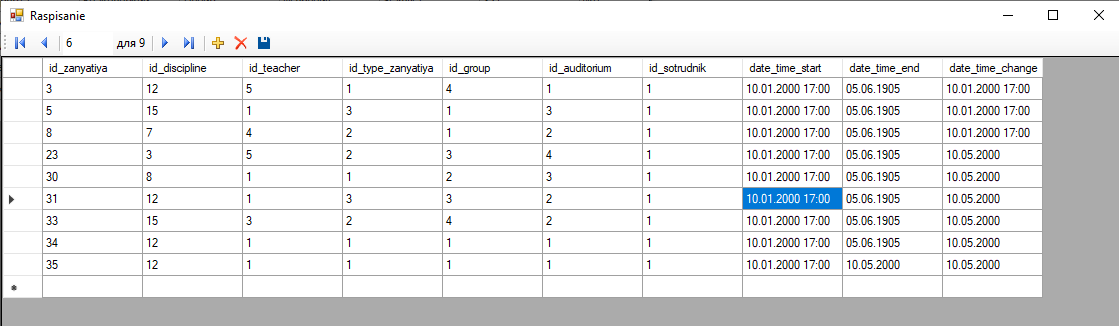


Рисунок 30 – Справочник расписание

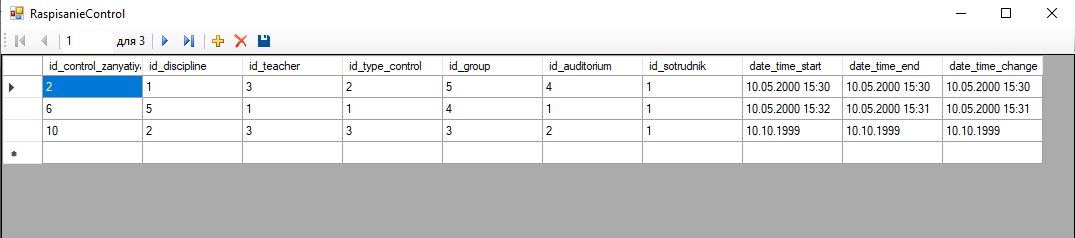


Рисунок 31 – Справочник расписание контроля

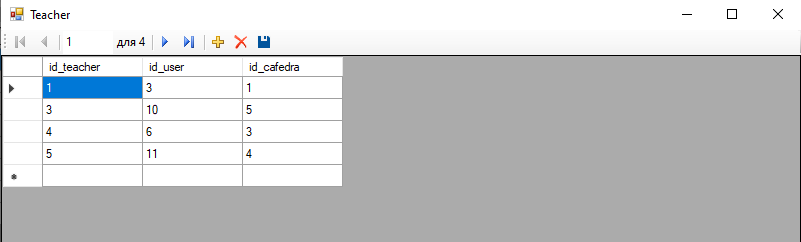


Рисунок 32 – Справочник преподаватели

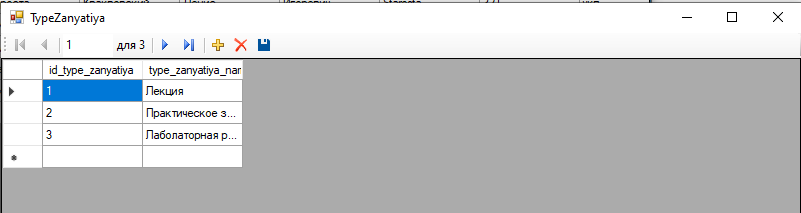


Рисунок 33 – Справочник тип занятия

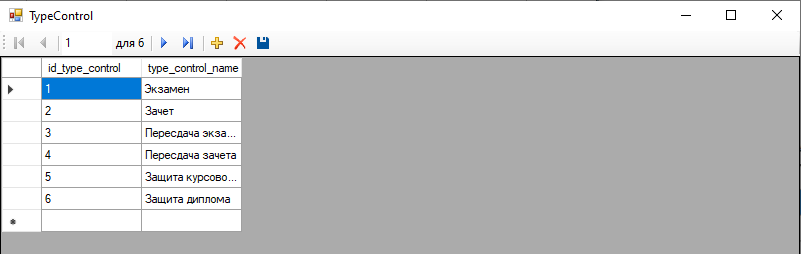


Рисунок 34 – Справочник тип контроля

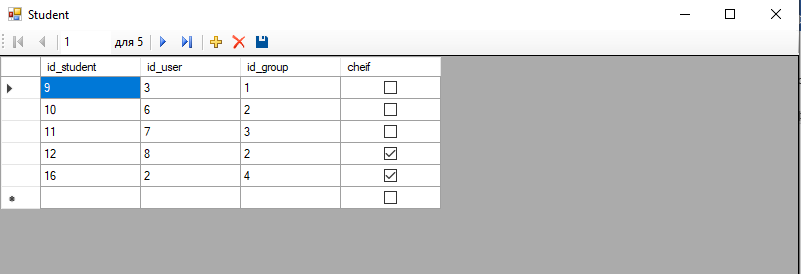


Рисунок 35 – Справочник студенты

При нажатии на кнопку Справка выводится пункт о программе как на рисунке 35:

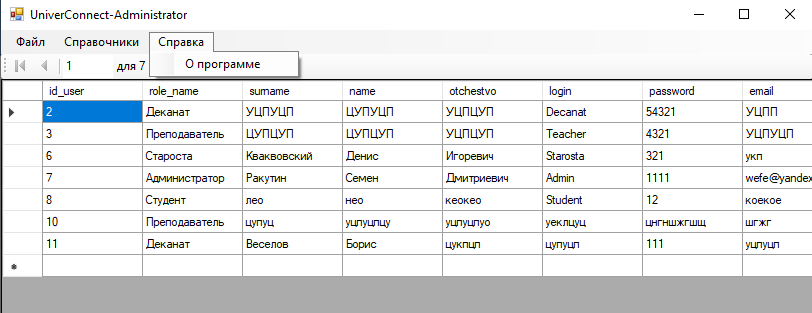


Рисунок 36 – Пункт Справка

В этом пункте содержится информация о программе и ее создателе, которую можно увидеть на рисунке 36:

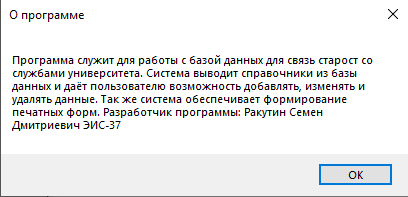


Рисунок 37 – О программе

Если мы хотим завершить работу, то нажимаем Выход в меню Файл, как показано на рисунке 37 и программа закрывается:

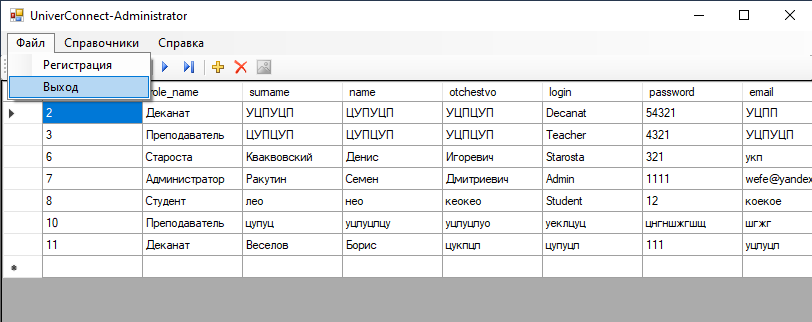


Рисунок 38 – Выход из программы

Когда пользователь авторизовался как сотрудник деканата ему доступен следующий функционал:

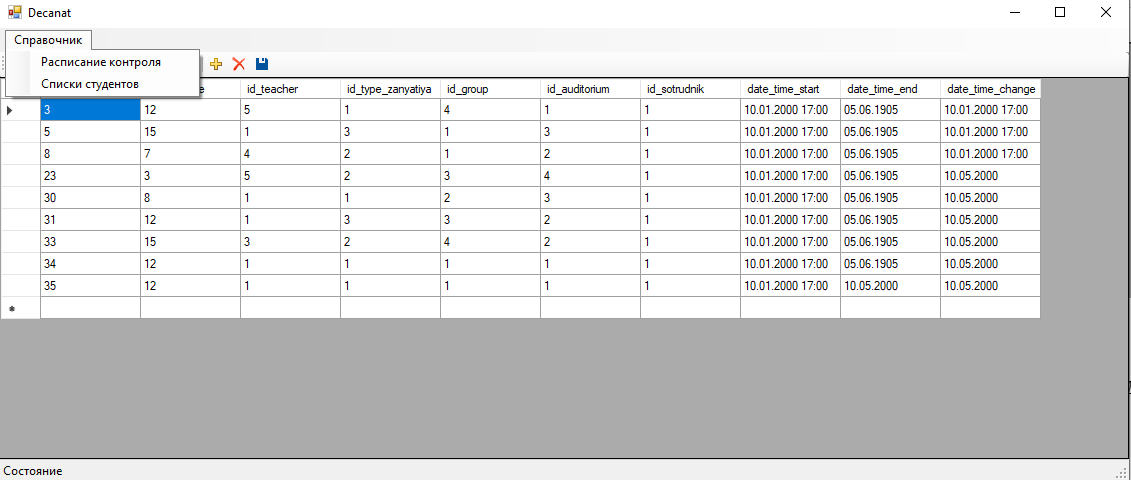


Рисунок 39 – выбор справочника

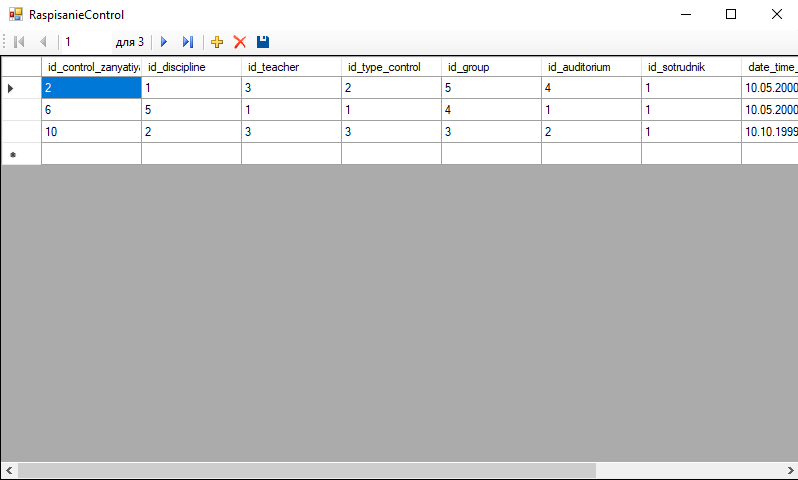


Рисунок 40 – справочник расписание контроля

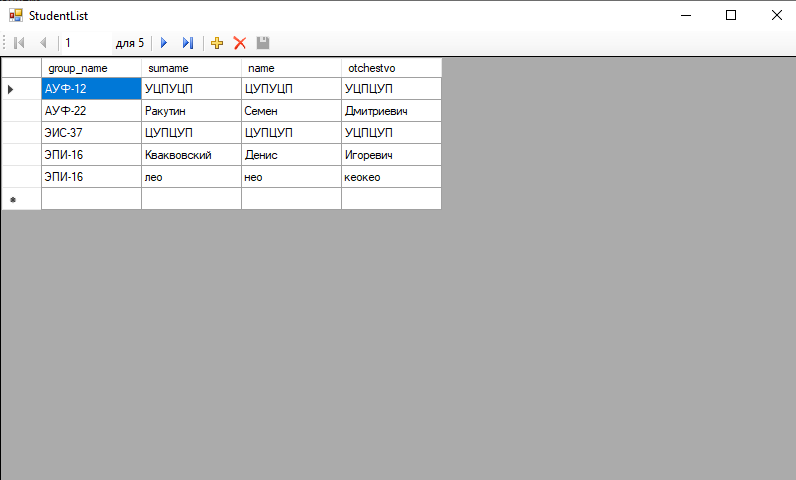


Рисунок 41 – справочник список студентов

Когда пользователь авторизовался как преподаватель ему доступен следующий функционал:

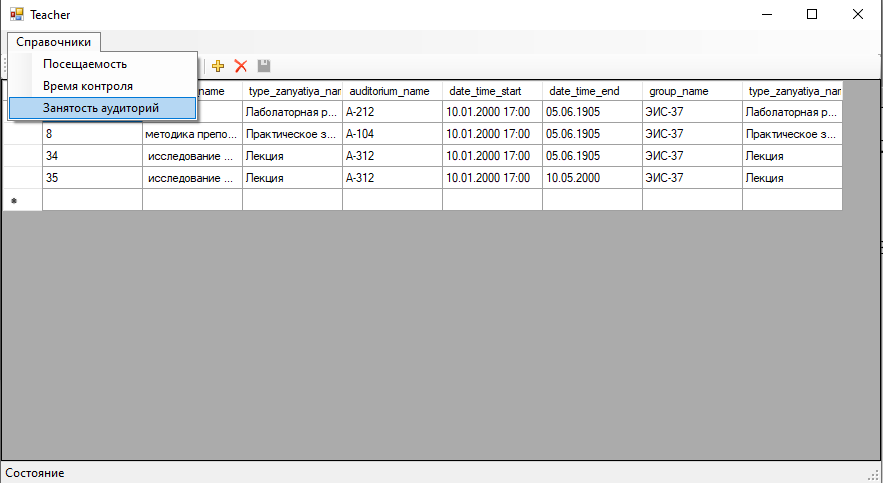


Рисунок 42 – выбор справочника

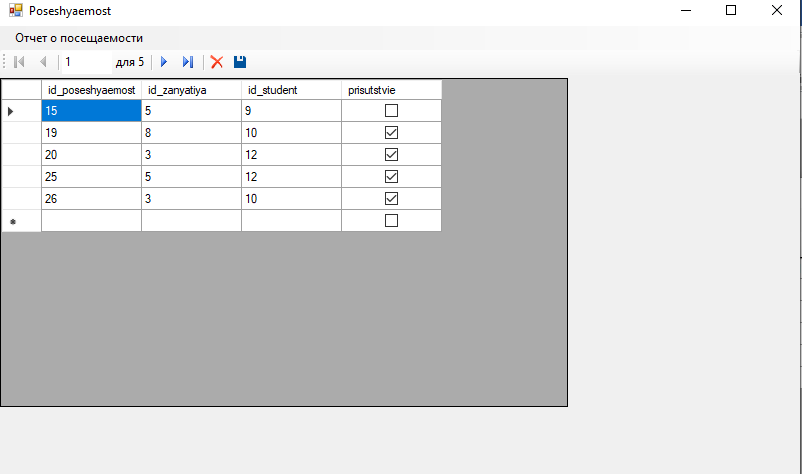


Рисунок 43 – справочник посещаемость

Если нажать кнопку “Печать отчета о посещаемости”, то соответственно открывается форма отчета о посещаемости как на рисунке 43 с возможностью сохранения отчета в формате Excel:

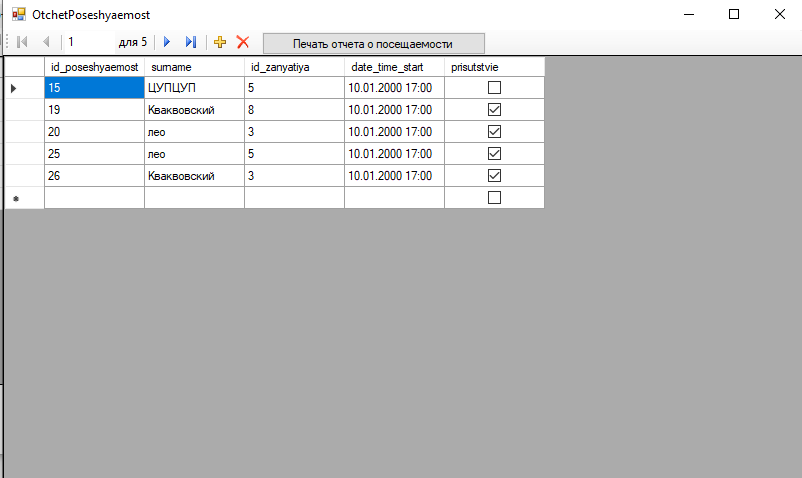


Рисунок 44 – отчет о посещаемости

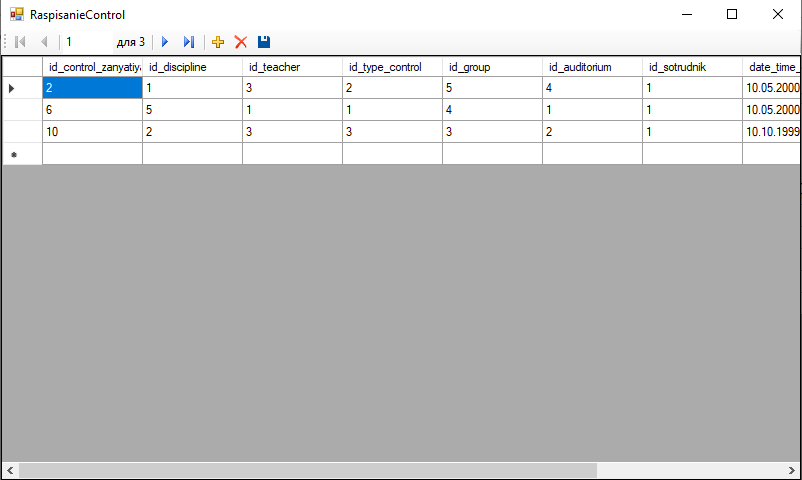


Рисунок 45 –Справочник расписание контроля

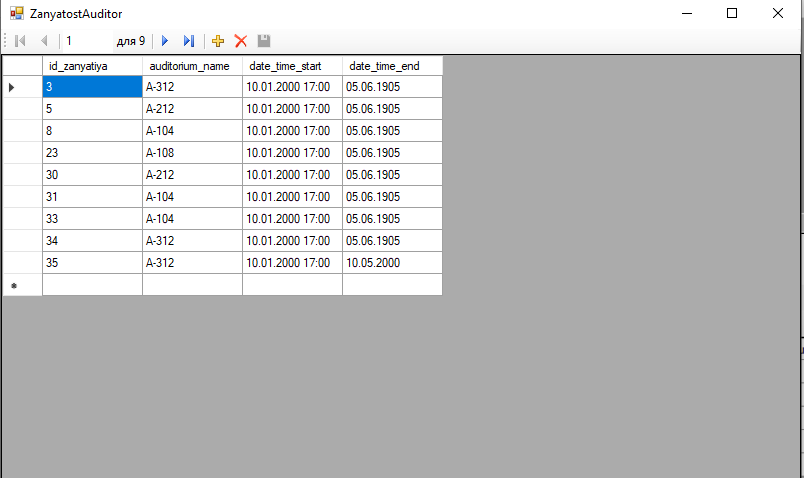


Рисунок 46 – справочник занятость аудиторий

Когда пользователь авторизовался как староста ему доступен следующий функционал:

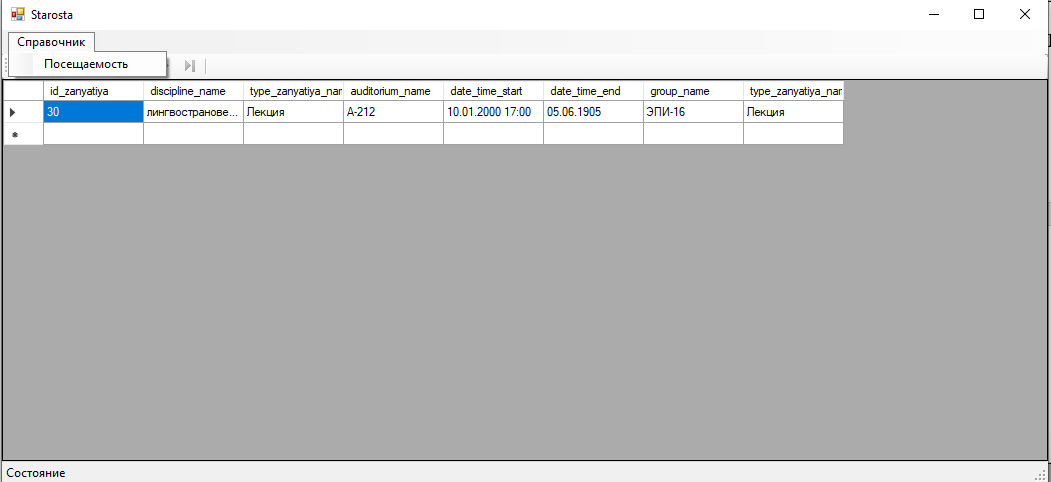


Рисунок 47 – выбор справочника

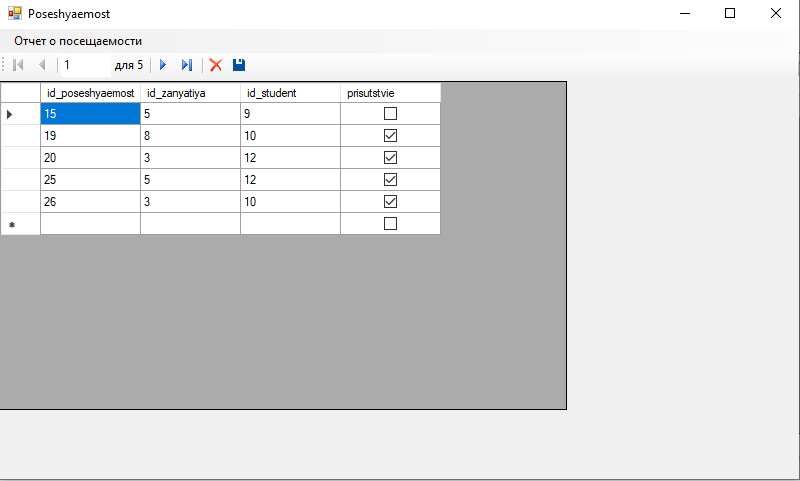


Рисунок 48 – справочник посещаемость

Если нажать кнопку “Печать отчета о посещаемости”, то соответственно открывается форма отчета о посещаемости как на рисунке 46 с возможностью сохранения отчета в формате Excel:

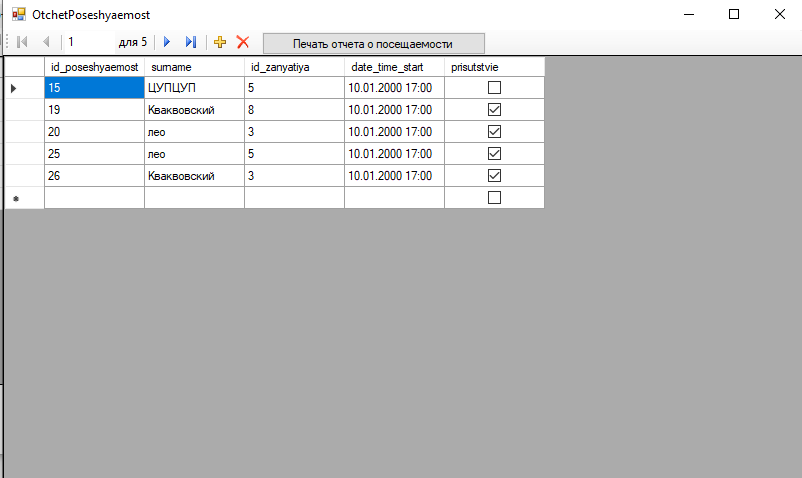


Рисунок 49 – отчет о посещаемости

Когда пользователь авторизовался как студент ему доступен следующий функционал:

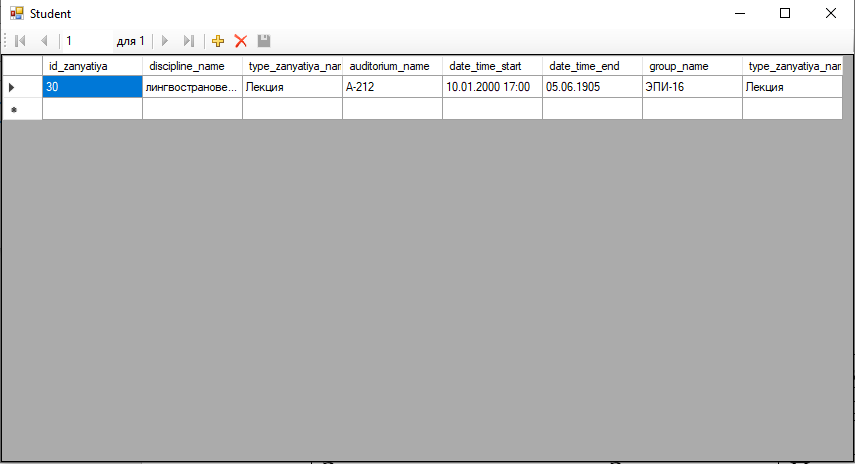


Рисунок 50 – авторизация как студент

# Описание тест – кейсов

|  |  |
| --- | --- |
| **Название** | **Проверка формы регистрации** |
| **Действия** | **Ожидаемый результат** |
| **Предусловие:** | |
| Запустить приложение. Запустить Файл Регистрация. | Приложение корректно запустилось. Открылось окно Регистрация. |
| **Проверка 1: Успешная авторизация пользователя с указанной ролью** | |
| 1.Ввести корректные данные в поля «Логин», «Пароль», «Пользователь», «Роль». | 1.Не вывелось никаких сообщений об ошибках. |
| 2.Нажать на кнопку «Регистрация». | 2.Форма регистрации закрылась, новый пользователь зарегистрирован. |
| **Проверка 2: Проверка работы при вводе некорректных данных** | |
| 1.После открытия формы не вводить ничего в поля «Логин», «Пароль», «Пользователь», «Роль». | 1.Открылось окно с сообщением «Заполнены не все поля». |
| 2.После открытия формы ввести корректные данные, но некоторое поля оставить пустыми. | 2.Открылось окно с сообщением «Заполнены не все поля». |
| 3.После открытия формы ввести корректные данные в поля «Логин», «Пароль», «Пользователь», но выбрать несуществующую роль. | 3.Открылось окно с сообщением «У вас недостаточно прав». |

Таблица 14 – Тест – кейс для проверки Регистрации

|  |  |
| --- | --- |
| **Название** | **Проверка редактирования данных в справочнике** |
| **Действия** | **Ожидаемый результат** |
| **Предусловие:** | |
| Открыть справочник. | Открылся справочник. |
| **Проверка 1: Успешное изменение, добавление или удаление данных справочника** | |
| 1.После открытия формы ввести данные в соответствующие поля. | 1.Не вывелось никаких сообщений об ошибках. |
| 2.Нажать на кнопку «Изменить»\ «Добавить». | 2.Данные в справочнике изменились\добавились. |
| 3.Нажать на кнопку «Удалить». | 3. Данные из справочника удалились. |
| 4.Нажать на кнопку «Ввод изменений в БД». | 4.После вывода сообщения о соглашении или несоглашении с изменениями, данные изменяются в БД. |
| **Проверка 2: Проверка работы при вводе некорректных данных** | |
| 1.После открытия формы не вводить ничего в соответствующие поля и сразу нажать кнопки «Изменить», «Добавить» или «Удалить». | 1.Вывелось окно с сообщением «Сохраните предыдущие изменения или заполните все поля». |
| 2.После открытия формы не вводить ничего в соответствующие поля и сразу нажать кнопку «Ввод данных в БД». | 2.Вывелось окно с сообщением «Не введено значение категории» |

Таблица 15 – Тест – кейс для проверки Справочника

# Заключение

В ходе выполнения курсового проекта мною были решены следующие пункты:

1. Проведен анализ необходимых функций и областей проекта
2. Создана модель данной информационной системы
3. Создана база данных дня данной информационной системы
4. Создано клиентское приложение на языке программирования С# в среде Microsoft Visual Studio 2019

В ходе выполнения курсового проекта я ознакомилась с методами установки соединения между приложением и сервером SQL Server, а также научилась создавать приложение для работы с базой данных, осуществлять администрирование и вывод данных в MS Excel документ.

# Список используемых источников и литературы

1. Ткачев О.А. Создание и манипулирование базами данных средствами СУБД Microsoft SQL Server 2008 [Электронный pecypc]: Учебное пособие/ Ткачев О.А. – Электрон. текстовые данные. – М.: Московский городской педагогический университет, 2013. – 152 с.     – Режим доступа: http://www.bibliocomlectator.ru/book/?id=26613
2. Чубукова И.А. Data Mining [Электронный pecypc]/ Чубукова И.А. – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий  (ИНТУИТ), 2016. – 470 с. – Режим доступа: <http://www.bib1iocomp1ectator.ru/book/?id–56315>
3. В.В.Соколова ПОДКЛЮЧЕНИЕ VISUALSTUDIO 2013 C# К БАЗЕ ДАННЫХ SQL\_\_\_\_\_ SERVER
4. Александр Бондарь Microsoft SQL Server 2014
5. Ицик Бен-Ган Microsoft SQL Server 2012. Основы T-SQL

# Приложение А.

# Приложение Б.

Примеры отчетов.

Как было описано ранее в созданном приложении есть возможность печати отчетов. Следующий код описывает сохранение отчета в документ Excel:

saveFileDialog1.Filter = "Файлы Excel (\*.xls; \*.xlsx) | \*.xls; \*.xlsx"; // Отображаютя только файлы Excel

if (saveFileDialog1.ShowDialog() == DialogResult.OK) // Если пользователь сохранил документ

{

// Создание Excel документа

Microsoft.Office.Interop.Excel.\_Application app = new Microsoft.Office.Interop.Excel.Application();

// Создание новой рабочей книги в этом документе

Microsoft.Office.Interop.Excel.\_Workbook workbook = app.Workbooks.Add(Type.Missing);

// Создание нового листа в вышесозданной книге

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

// Устанавливает свойство видимости документа за программой. Можно установить false

app.Visible = true;

worksheet = workbook.ActiveSheet; // Определение значения объекта

worksheet.Name = "Отчет Посещаемость"; // Изменение имени рабочего листа

// Заполнение Excel документа

worksheet.Cells[1, 1] = "Отчет Посещаемость:";

worksheet.Cells[2, 1] = "id посещаемости";

worksheet.Cells[2, 2] = "Фамилия студента";

worksheet.Cells[2, 3] = "id занятия";

worksheet.Cells[2, 4] = "Дата и время занятия";

worksheet.Cells[2, 5] = "Присутствие";

for (int i = 1; i < poseshaemostDataGridView.Columns.Count + 1; i++)

{

worksheet.Columns[i].ColumnWidth = 30;//Установление ширины столбцов

worksheet.Cells[3, i].Font.Color = Color.Red;//Установление цвета шрифта столбцов

worksheet.Cells[3, i] = poseshaemostDataGridView[i - 1, 0].Value;

}

for (int i = 1; i < poseshaemostDataGridView.RowCount; i++)

for (int j = 0; j < poseshaemostDataGridView.ColumnCount; j++)

{ worksheet.Cells[i + 3, j + 1] = poseshaemostDataGridView[j, i].Value; }

// Сохраняет документ

workbook.SaveAs(saveFileDialog1.FileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

app.Quit(); // Закрывает документ

На рисунке 49 показан отчет, выгруженный в программу просмотра и редактирования табличных документов Microsoft Excel:

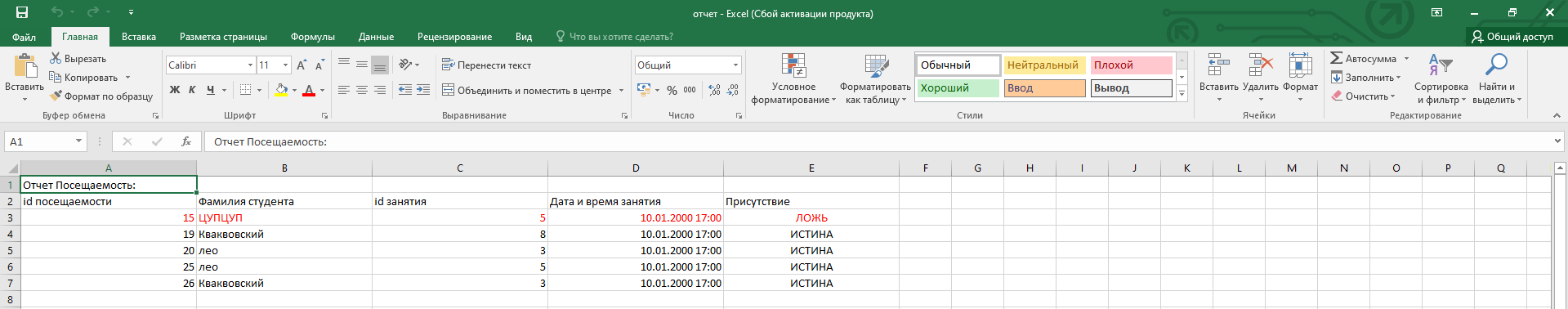


Рисунок 51 – Отчет о посещаемости в формате Excel

# Приложение В

Листинг наиболее важных частей кода

Код, описывающий подключение к БД:

// Создание строки подключения

String ConnectString = "Integrated Security=true; User Id = " + textBox1.Text + "; Password = " + textBox2.Text + @"; server= " + textBox3.Text + "; Initial Catalog = " + textBox4.Text;

// Объявление новой переменной типа SqlConnection

SqlConnection con = new SqlConnection(ConnectString);

Exception error = null; // Переменная, представляющая ошибки, появляющиеся во время выполнения приложения

try

{

con.Open(); // Открытие подключения

}

catch (Exception ex) // При возникновении неполадок при подключении появится сообщение с информацией об ошибке

{

error = ex; // Переменная error запоминает конкретную ошибку

MessageBox.Show(string.Format("Error: {0}", ex.Message), "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

if (error == null) // Если переменная не изменилась, то ошибок при подключении не было и соединение с БД установлено

{

SqlCommand command = new SqlCommand($" sp\_helpuser'{textBox1.Text}'", con);

SqlDataReader reader = command.ExecuteReader();

reader.Read();

string a = reader["Rolename"].ToString();

reader.Close();

// con.Close();

// определяю роль текущего пользователя

SqlCommand command1 = new SqlCommand($" SELECT id\_user FROM dbo.\_User WHERE(login = SUSER\_SNAME())", con);

SqlDataReader reader1 = command1.ExecuteReader();

reader1.Read();

int id\_user = Convert.ToInt32(reader1[0]);

reader1.Close();

// определяю id текущего пользователя

SqlCommand command2 = new SqlCommand($"SELECT DISTINCT Changed.id\_user FROM Changed where Changed.id\_user = '{id\_user.ToString()}' ", con);

SqlDataReader reader2 = command2.ExecuteReader();

reader2.Read();

bool est = reader2.HasRows;

reader2.Close();

con.Close();

switch (a)

{

case "db\_owner":

MDIParent1 mdi = new MDIParent1(con);

mdi.Show();

break;

case "DecanatRole": Decanat decanat = new Decanat();

decanat.Show(); break;

case "TeacherRole": Teacher teach = new Teacher(id\_user,est,con);

teach.Show();

break;

case "StarostaRole":

Starosta starosta = new Starosta(con, id\_user,est);

starosta.Show();

break;

case "StudentRole": Student stud = new Student(id\_user,est,con);

stud.Show();

break;

}

con.Close();

Код регистрации нового пользователя:

try

{

if (textBox1.Text != null && textBox2.Text != null && textBox3.Text != null && comboBox1.SelectedIndex != -1)

{

int role = comboBox1.SelectedIndex;

string str = "";

switch (role)

{

case 0:

str = "db\_owner";

break;

case 1:

str = "DecanatRole";

break;

case 2:

str = "TeacherRole";

break;

case 3:

str = "StarostaRole";

break;

case 4:

str = "StudentRole";

break;

}

role++;

String quertString = @"insert into \_User (login, password, id\_role) values ('" + textBox2.Text + "','" + textBox3.Text + "','" + role.ToString() + "');";

SqlCommand insert = new SqlCommand(quertString, con);

con.Open();

insert.ExecuteNonQuery();

con.Close();

//СОЗДАНИЕ

String quertString2 = @"EXEC sp\_addlogin " + textBox2.Text + "," + textBox3.Text + ",UniverConnect";

SqlCommand create2 = new SqlCommand(quertString2, con);

con.Open();

create2.ExecuteNonQuery();

con.Close();

String quertString3 = @"EXEC sp\_adduser " + textBox2.Text + "," + textBox1.Text + "";

SqlCommand create3 = new SqlCommand(quertString3, con);

con.Open();

create3.ExecuteNonQuery();

con.Close();

String quertString4 = @"EXEC sp\_addrolemember " + str + "," + textBox1.Text + " ";

SqlCommand create4 = new SqlCommand(quertString4, con);

con.Open();

create4.ExecuteNonQuery();

con.Close();

MDIParent1 my = Application.OpenForms.OfType<MDIParent1>().Single();

my.UserRegistered();

}

else

{

MessageBox.Show("Заполнены не все поля", "Ошибка");

}

this.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "У вас недостаточно прав!");

}