|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУ-КФ «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_***ИУ5-КФ «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту на тему:**

**«**Электронный журнал»

по дисциплине ***Базы данных***

Студент гр. САПР Б-51 (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) Давыдов А.С.

(подпись) (Ф.И.О.)

Руководитель (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) Кириллов В.Ю.

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2020

***Министерство науки и высшего образования Российской Федерации***

*Калужский филиал  
 федерального государственного бюджетного   
образовательного учреждения высшего образования*

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***

***(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУ5-КФ\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(Е.В.Вершинин)

«\_\_»\_\_\_\_сентября\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

по дисциплине \_\_\_***Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент Давыдов А.С. САПР.Б-51\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель Кириллов В.Ю.

График выполнения проекта: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

1. Тема курсового проекта

Электронный журнал

2. Техническое задание

Изучить предметную область, а именно заполнение журналов посещаемости и успеваемости. Разработать работающую систему для упрощения работы преподавателя.

***3. Оформление курсового проекта***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)

Презентация по данному проекту

Дата выдачи задания «\_\_8\_\_\_»\_\_сентября\_2019г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/В.Ю.Кириллов/

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«5»\_\_сентября\_2019г

**Содержание**

[1.Введение 4](#_Toc35635166)

[2.Техническое задание 4](#_Toc35635167)

[2.1 Общие сведения 4](#_Toc35635168)

[2.1.1 Наименование 4](#_Toc35635169)

[2.1.2 Заказчик и разработчик 4](#_Toc35635170)

[2.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ: 4](#_Toc35635171)

[2.1.4 Плановые сроки по созданию приложения 4](#_Toc35635172)

[2.1.5 Порядок оформления и предъявления заказчику результатов работ 4](#_Toc35635173)

[2.2 Назначение и цели создания системы 5](#_Toc35635174)

[2.2.1 Назначение системы 5](#_Toc35635175)

[2.2.2 Цели создания системы 5](#_Toc35635176)

[2.3 Требования к системе 5](#_Toc35635177)

[2.3.1 Требования к системе в целом 5](#_Toc35635178)

[2.3.2 Требования к надежности 5](#_Toc35635179)

[2.3.3 Требования к функциям, выполняемым системой 5](#_Toc35635180)

[2.3.4 Требования к информационному обеспечению системы 5](#_Toc35635181)

[2.3.5 Требования к методическому обеспечению 6](#_Toc35635182)

[2.5 Состав и содержание работ по созданию(развитию) системы 6](#_Toc35635183)

[2.6 Порядок контроля и приемки системы 6](#_Toc35635184)

[2.6.1 Методы испытания системы 6](#_Toc35635185)

[2.6.2 Общие требования к приемке работ по стадиям 6](#_Toc35635186)

[2.6.3 Статус приемочной комиссии 6](#_Toc35635187)

[3.Исследовательская часть 8](#_Toc35635188)

[3.1. Постановка задачи 8](#_Toc35635189)

[3.2. Описание предметной области 8](#_Toc35635190)

[3.2.1 Общие сведения 8](#_Toc35635191)

[3.2.2. Перечень реализуемых функций 8](#_Toc35635192)

[3.3 Сценарии использования 9](#_Toc35635193)

[3. Перечень задач, подлежащих решению в процессе разработки 9](#_Toc35635194)

[4.Проектно-Конструкторская часть 10](#_Toc35635195)

[4.1 Процесс проектирования приложения 10](#_Toc35635196)

[4.2 Логическое проектирование. Нормальные формы. 14](#_Toc35635197)

[4.3 Логическая схема базы данныx 16](#_Toc35635198)

[4.4. Разработка интерфейса взаимодействия пользователя с системой. 18](#_Toc35635199)

[5.Проектно-Технологическая часть 20](#_Toc35635200)

[5.1. Порядок развертывания системы. 20](#_Toc35635201)

[5.2. Назначение системы 20](#_Toc35635202)

[5.3 Уровень подготовки пользователя 20](#_Toc35635203)

[Заключение 21](#_Toc35635204)

[Список использованных источников 22](#_Toc35635205)

[Листинг 23](#_Toc35635206)

# 1.Введение

Цель данной курсовой работы является разработка электронного журнала студента. Из данной цели вытекают следующие задачи:

1. Обеспечить возможность следить за успеваемостью и посещаемостью студентов в режиме реального времени.
2. Обеспечить преподавателям возможность предоставления информации об успеваемости.

3) Обеспечить студентам самостоятельный контроль своей успеваемости в течении семестра.

# 2.Техническое задание

## **2.1 Общие сведения**

### 2.1.1 Наименование

“Электронный журнал занятий”

### 2.1.2 Заказчик и разработчик

Заказчик: Кириллов В.Ю.

Почта: [sapr91@mail.ru](mailto:sapr91@mail.ru)

Разработчик: Давыдов А.С.

Почта: lxdvd1999@gmail.com

### 2.1.3 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:

-Методические указания по выполнению курсовой работы по дисциплине “Базы данных”, автором которых является В.Ю. Кириллов.

-А также на основании Технического задания .

### 2.1.4 Плановые сроки по созданию приложения

Начало работ – 3 сентября 2019 года, окончание работ -1 декабря 2019 года.

# 2.1.5 Порядок оформления и предъявления заказчику результатов работ

Работы по созданию Электронного журнала занятий сдаются Разработчиком поэтапно в соответствии с календарным планом Проекта. По окончанию каждого этапа Разработчик сдает Заказчику соответствующие отчетные документы этапа.

## **2.2 Назначение и цели создания системы**

### 2.2.1 Назначение системы

Электронный журнал занятий предназначен для того, что бы предоставлять пользователям сведения об успеваемости и посещаемости.

### 2.2.2 Цели создания системы

Данное приложение создается с целью возможности заменить бумажные дневники и журналы.

## **2.3 Требования к системе**

### 2.3.1 Требования к системе в целом

СУБД: Microsoft SQL Server 2008 R2, язык высокого уровня: C# в среде Microsoft Visual Studio 2015.

### 2.3.2 Требования к надежности

В случае, если любой пользователь приложения совершает ошибку при его использовании, он должен быть уведомлен об этом, причем приложение должно оставаться работоспособным.

### 2.3.3 Требования к функциям, выполняемым системой

Приложение должно обеспечивать реализацию следующих функций:

* Формирование списка студентов, изучающих дисциплину;
* Определение преподавателя, ведущего занятия;
* Проверка полномочий преподавателя;
* Учет посещения занятий;
* Учет сданных работ;
* Подсчет количества пропущенных часов;
* Формирование списка студентов, имеющих задолженность по заданиям.

### 2.3.4 Требования к информационному обеспечению системы

Хранение и обслуживание данных в приложении должно происходить

с помощью одной из существующих систем управления базами данных.

Информация должна сохранять целостность, не должно возникать

аномалий при попытке удаления, изменения и добавления данных.

### 2.3.5 Требования к методическому обеспечению

Разработку системы требуется сопровождать расчетно-пояснительной

запиской и графической частью. Расчетно-пояснительная записка состоит

из следующих частей:

1. Техническое задание

2. Исследовательская часть. (Научно-исследовательская часть.)

3.Проектно - конструкторская часть.

4. Проектно-технологическая часть.

## **2.5 Состав и содержание работ по созданию(развитию) системы**

Первый этап – изучение предметной области, определение состава и структуры данных. В результате прохождения первого этапа определяются: логическая структура данных, ТЗ , описание объекта автоматизации и целей разработки.

Второй этап – анализ пользовательских сценариев работы. В результате создается пользовательский интерфейс.

Третий этап – анализ взаимодействия веб-приложения с серверной компонентой. В результате определяются: архитектура приложения и структура классов.

## **2.6 Порядок контроля и приемки системы**

### 2.6.1 Методы испытания системы

Испытания проектируемой системы проводятся заказчиком.

### 2.6.2 Общие требования к приемке работ по стадиям

Место проведения приемки работ КФ МГТУ им. Н.Э.Баумана.

Сроки проведения приёмки работ – 7 октября, 21 октября, 4 ноября, 18 ноября, 2 декабря и другие даты согласованные с заказчиком и исполнителем

### 2.6.3 Статус приемочной комиссии

Прием работы на всех этапах осуществляется непосредственным руководителем курсовой работы. По окончанию разработки работоспособное приложение демонстрируется на комиссии.

**2.7 Требования к документированию**

-ТЗ по проектируемой системе

-Расчетно-пояснительная записка

-Презентация Microsoft PowerPoint.

# 3.Исследовательская часть

## **3.1. Постановка задачи**

Требуется разработать приложение для учета посещаемости и успеваемости студентов.

## **3.2. Описание предметной области**

### 3.2.1 Общие сведения

Основной задачей приложения является учет посещаемости и успеваемости студентов.

В системе используются 4 сущности: электронный журнал, студент, преподаватель и дисциплина. Каждый студент имеет свой уникальный код, ФИО, дату рождения и пол. Каждый преподаватель имеет так же свой уникальный код, ФИО и полномочия. Дисциплина включает в себя код, название, количество занятий, количество лабораторных работ. Сущность электронный журнал включает в себя код дисциплины, код занятия, оценку, код студента и код преподавателя.

В приложение могут заходить как преподаватели, так и студенты. Преподаватель выполняет заполнение журнала, учет посещаемости и сдачи работ и выставление оценок за сдачу. Студент в ,свою очередь, может выполнить вход в приложение и просмотреть оценки и свою посещаемость.

3.2.2. Перечень реализуемых функций

* Формирование списка студентов, изучающих дисциплину;
* Определение преподавателя, ведущего занятия;
* Проверка полномочий преподавателя;
* Учет посещения занятий;
* Учет сданных работ;
* Подсчет количества пропущенных часов;
* Формирование списка студентов, имеющих задолженность по заданиям.

## **3.3 Сценарии использования**

Студент:

Войдя в приложение, студент может ,в большинстве случаев, ознакомится с предоставляемой информацией.

1)Просмотр посещаемости.

2)Просмотр успеваемости и оценок за сдачу лабораторных работ.

Преподаватель:

1) Преподаватель ведет таблицу посещаемости студентов на каждом проведенном занятии.

3) Так же преподаватель отмечает и оценивает сдачу лабораторных работ по дисциплине.

3) Если необходимо преподаватель формирует список сдачи лабораторных работ и посещаемости.

## **3. Перечень задач, подлежащих решению в процессе разработки**

Для создания системы необходимо:

1.Составить техническое задание.

2.Изучить предметную область и на основе полученной информацию определить структуру данных, необходимую для выполнения поставленных задач.

3.Разработать план поэтапного формирования , так как реализация всего описанного не представляется возможной в пределах данной курсовой работы.

4. Изучить состав и структуру данных.

5.Создать или изучить готовые алгоритмы необходимых операций.

6. Определить взаимодействия между базой данных и приложением.

7. Создать интерфейс взаимодействия пользователя с приложением.

## 

# 4.Проектно-Конструкторская часть

## **4.1 Процесс проектирования приложения**

Процесс проектирования включает в себя следующие этапы:

* инфологическое проектирование;
* определение требований к операционной среде, в которой будет функционировать информационная система, и выбор СУБД и инструментальных программных средств;
* логическое проектирование БД;
* физическое проектирование БД.

Основная цель процесса проектирования базы данных состоит в получении моделей БД, которые удовлетворяют следующим требованиям:

* корректность схемы БД, т.е. база должна адекватно отображать моделируемую предметную область, в которой каждому объекту предметной области соответствуют данные в структуре БД, а каждому процессу — процедуры обработки данных:
* обеспечение ограничений на целостность данных;
* эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных);
* простота и удобство эксплуатации;
* гибкость, т.е. возможность развития БД и адаптации к изменениям предметной области и/или требований пользователей.

**Инфологическая модель** предметной области представляет собой описание её структуры и динамики, характера информационных потребностей

пользователей в терминах, понятных пользователю и не зависимых от

реализации БД. Одним из наиболее часто используемых видов моделей является модель «сущность-связь» (ER-модель). Основными элементами этой модели являются сущность, атрибут и связь.

**Сущность** — это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие, так и абстрактные.

Сущности:

* 1. Электронный журнал (Абстрактная сущность)
  2. Преподаватель (Физическая сущность)
  3. Дисциплина (Абстрактная сущность)
  4. Студент (Физическая сущность)

Выделяют сильные и слабые типы сущностей. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. В данном случае в качестве независимой сущности выступают Преподаватель, Студент и Дисциплина, а в качестве подчиненной сущности – Электронный журнал.

Для каждой сущности выбираются свойства — атрибуты.

1. Электронный журнал

* Код дисциплины
* Код занятия/лабораторной работы
* Оценка
* Код преподавателя
* Код студента

1. Преподаватель

* Код преподавателя
* Имя преподавателя
* Полномочия

1. Дисциплина

* Код дисциплины
* Название дисциплины
* Добавлен ли соответствующий журнал
* Количество занятий
* Количество лабораторных работ

1. Студент

* Номер (код) студента
* Имя студента
* Дата рождения
* Пол

Различают следующие атрибуты:

1. *идентифицирующие и описательные атрибуты.* Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ, включающий в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и заключают в себе интересующие свойства сущности. *Для сущности Преподаватель идентифицирующими будут являться код преподавателя и код дисциплины, а имя и полномочия описательными, для сущности Студент идентифицирующие – код студента и код дисциплины, а описательными – имя, дата рождения и пол, для сущности Дисциплина идентифицирующие – код дисциплины, а описательные – количество занятий и лабораторных работ и добавление в журнал, для сущности Электронный журнал – идентифицирующие код дисциплины, а описательными – код занятия/ лабораторной работы, оценка, код преподавателя и студента* ;
2. *составные и простые атрибуты.* Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута. *В данном случае такие атрибуты как код дисциплины, занятия, преподавателя, студента оценка, пол, добавление в журнал и количество занятий или лабораторных работ, дата рождения являются простыми атрибутами, а имя преподавателя, студента, название дисциплины являются составными.* ;
3. *однозначные и многозначные атрибуты* (могут иметь соответственно одно или много значений для каждого экземпляра сущности);
4. *основные и производные атрибуты.* Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

По типу связи различают следующие виды:

* *связи один к одному*

Каждой записи первой сущности соответствует только одна запись из второй сущности. А каждой записи второй сущности соответствует только одна запись из первой сущности. *Данным типом связи атрибуты cod\_pr в сущности ERegister и cod\_pr в сущности Teacher, cod\_ds в сущности ERegister и cod\_ds в сущности Discplines, cod\_ds в сущности Teacher и cod\_ds в сущности Discplines, cod\_ds в сущности Student и cod\_ds в сущности Discplines.*

* *один ко многим*

Каждой записи первой сущности могут соответствовать несколько записей из второй сущности. Однако каждой записи второй сущности соответствует только одна запись из первой сущности. *Данным типом связи связаны атрибут polnomochiya в сущности Teacher с атрибутами cod\_z, mark в сущности ERegister;*

* *многие ко многим*

Каждой записи первой сущности могут соответствовать несколько записей из второй сущности. Однако каждой записи второй сущности соответствует только одна запись из первой сущности.

## **4.2 Логическое проектирование. Нормальные формы.**

Ha этапе логического проектирования разрабатывается логическая структура базы данных. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД.

Проектирование реляционной базы данных (РБД) имеет свои особенности. Оно должно отвечать дополнительным требованиям, связанным с минимизацией дублирования данных и упрощения процедур их обработки и обновления. В случае неправильно спроектированной РБД могут возникнуть аномалии модификации данных, которые обусловлены недостаточно полно проведенным анализом типов связей между объектами предметной области и неполнотой анализа правил целостности данных. Для устранения таких недостатков применяется процедура нормализации отношений, образующих схему РБД, и определяются правила целостности базы данных.

Для определения понятия нормальной формы было введено понятие простого и сложного атрибута. Простой атрибут — это атрибут, значения которого атомарны (т.е. неделимы с точки зрения их обработки). Сложный атрибут может иметь значение, представляющее собой объединение нескольких значений, каждое из которых может обрабатываться самостоятельно. Примером сложного атрибута может быть агрегат данных или повторяющийся агрегат данных.

**Первая нормальная форма**

Чтобы преобразовать сущность в первую нормальную форму, следует исключить повторяющиеся группы значений и добиться того, чтобы каждый атрибут содержал только одно значение, списки значений не допускаются.  
Другими словами, каждый атрибут в сущности должен храниться только в одном экземпляре.

**Вторая нормальная форма**

Таблица во второй нормальной форме содержит только те данные, которые к ней относятся. Значения не ключевых атрибутов сущности зависят от первичного ключа. Если более точно, то атрибуты зависят от первичного ключа, от всего первичного ключа и только от первичного ключа.  
Для соответствия второй нормальной форме сущности должны быть в первой нормальной форме.

**Третья нормальная форма**

В третьей нормальной форме исключаются атрибуты, не зависящие от всего ключа. Любая сущность, находящаяся в третьей нормальной форме, находится также и во второй. Это самая распространенная форма базы данных.  
В третьей нормальной форме каждый атрибут зависит от ключа, от всего ключа и ни от чего, кроме ключа.

Все сущности, которые используются находятся уже в третьей нормальной форме, так как все атрибуты хранятся только в одном экземпляре, зависят от первичного ключа.

**4.3 Логическая схема базы данныx**

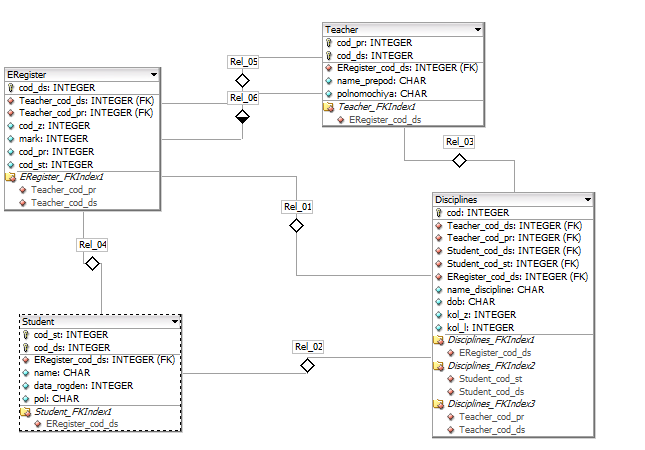


Рисунок 1 – Логическая схема создана в программе DBDesigner.

**Таблица ERegister:**

name\_ds: Код дисциплины

cod\_z: Код занятия/лабораторной работы

mark: Оценка

cod\_pr: Код преподавателя

cod\_st: Код студента

**Таблица Students:**

cod\_st: Номер (код) студента

name\_st: Имя студента

data\_rogden: Дата рождения

pol: Пол

**Таблица Teachers:**

cod\_pr: Код преподавателя

name\_prepod: Имя преподавателя

polnomochiya: Полномочия

**Таблица Discipline:**

cod\_ds: Код дисциплины

name\_discipline: Название дисциплины

dob: Добавлен ли соответствующий журнал

kol\_z: Количество занятий

kol\_l: Количество лабораторных работ

## **4.4. Разработка интерфейса взаимодействия пользователя с системой.**

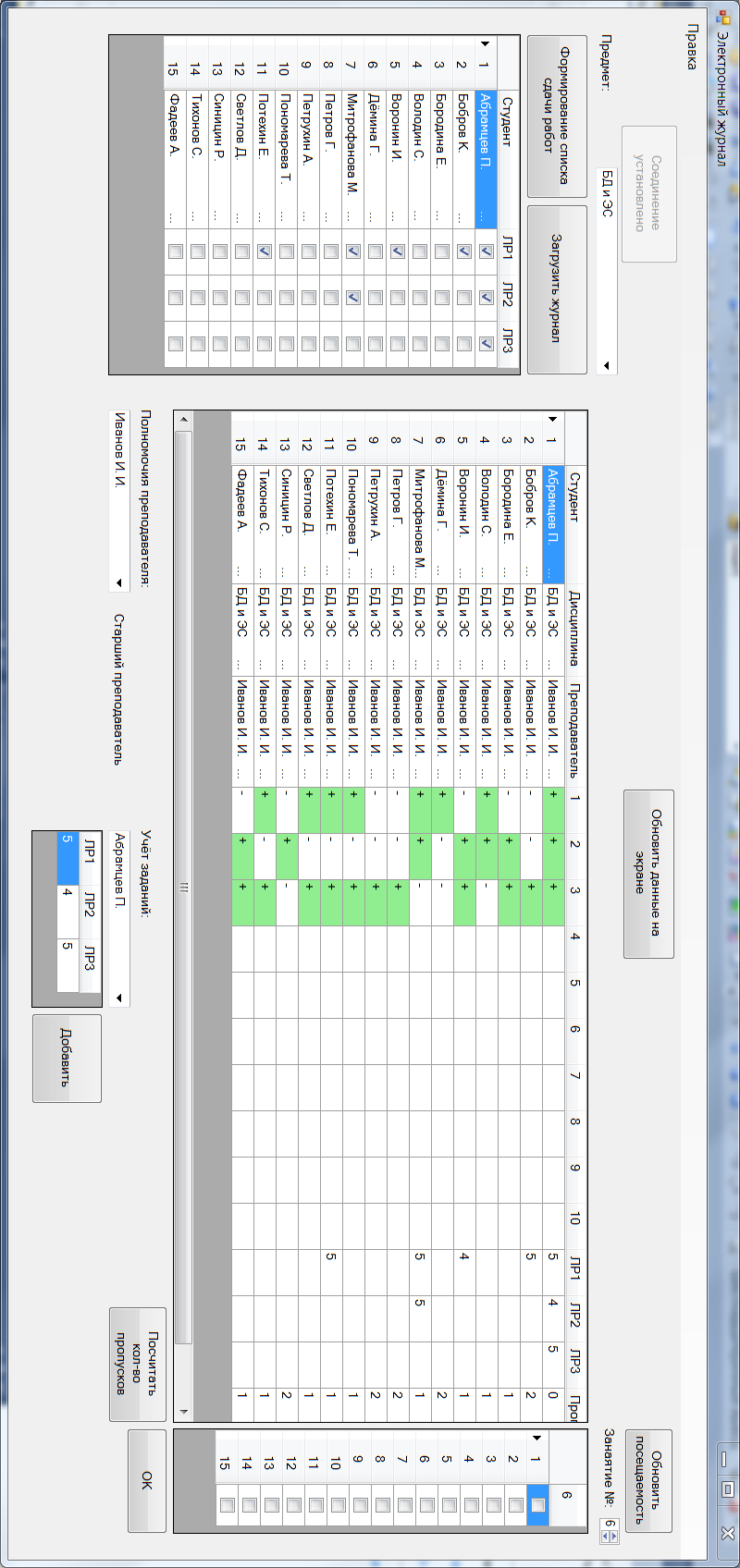


Рисунок 2 – Пример использования электронного журнала

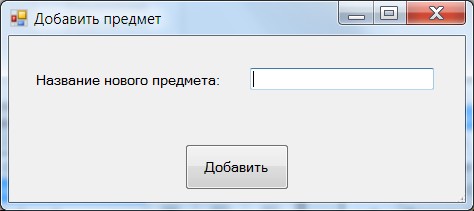


Рисунок 3 – Добавление нового предмета

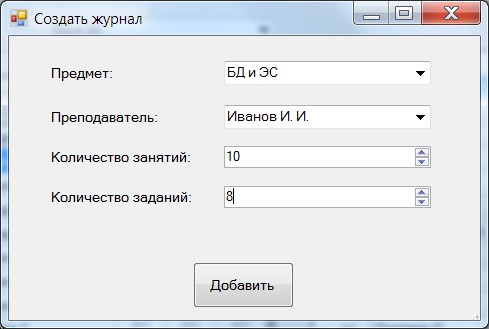


Рисунок 4 – Создание журнала занятий

# 5.Проектно-Технологическая часть

## **5.1. Порядок развертывания системы.**

Предполагаемый порядок развертывания системы состоит в последовательном выполнении следующих действий:

1. Установка соединения с базой данных (Метод SqlConnection.Open - Открывает подключение к базе данных со значениями свойств, определяемыми объектом ConnectionString)
2. Загрузка данных для заполнения журнала (Метод SqlDataReader.Read - Перемещает SqlDataReader к следующей записи. Метод SqlDataReader.Close - Закрывает объект SqlDataReader.)
3. Разработка оконного приложения, написание кода.

* Формирование таблицы задолженностей по лабораторным работам
* Учёт посещаемости
* Подсчет количества пропусков
* Добавление нового предмета
* Создание журнала по предмету

После запуска приложения пользователь может вести работу с журналом.

## **5.2. Назначение системы**

Система будет обеспечивать возможность учета посещаемости и сданных работ, а также хранить информацию об успеваемости.

## **5.3 Уровень подготовки пользователя**

Пользователь приложения должен иметь опыт работы с приложениями подобной структуры.

# Заключение

При выполнении курсовой работы на тему “Электронный журнал” была исследована и описана предметная область, а также обоснован выбор инструментов и платформы для разработки.

Также была разработана и реализована структура базы данных.

Результатом данной курсовой работы является рабочее приложение.

# Список использованных источников

Электронные ресурсы:

1. Руководство по программированию на C# [Электронный ресурс]-
2. Режим доступа:https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/
3. . Народный справочник [Электронный ресурс] - <https://ru.stackoverflow.com/>
4. Документация C# [Электронный ресурс] - <https://docs.microsoft.com/ru-ru/dotnet/csharp/>
5. Онлайн книга Изучаем ASP.NET MVC [Электронный ресурс]-
6. Режим доступа:https://metanit.com/sharp/mvc

# Листинг

**Используемые классы:**

**• SqlConnection - Предоставляет открытое подключение к базе данных SQL Server.**

**o Конструктор SqlConnection(string connectionString)**

**• SqlDataReader - Предоставляет возможность чтения потока строк только в прямом направлении из базы данных SQL Server.**

**• SqlCommand - Представляет инструкцию Transact-SQL или хранимую процедуру, выполняемую над базой данных SQL Server.**

**o Конструктор - SqlCommand(string cmdText, SqlConnection connection)**

**• SqlParametr - Предоставляет параметр для объекта SqlCommand.**

**o Конструктор - SqlParameter(string parameterName, SqlDbType dbType)**

**Установка соединения с базой данных:**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**myConnection = new SqlConnection("server=1-ПК;" + "Trusted\_Connection=yes;" + "database=master;" + "connection timeout=30");**

**try**

**{**

**myConnection.Open();**

**label1.Text = "Соединение установлено"; button1.Enabled = false;**

**}**

**catch (Exception exp)**

**{**

**label1.Text = exp.ToString();**

**}**

**}**

**Метод SqlConnection.Open - Открывает подключение к базе данных со значениями свойств, определяемыми объектом ConnectionString.**

**Загрузка данных для заполнения журнала на экране данными:**

**private void FillTheTable1()**

**{**

**string tab = comboBox3.Text; try**

**{**

**SqlDataReader myReader = null;**

**SqlCommand myCommand = new SqlCommand("select kol\_z, kol\_l from discipline where name\_discip='" + tab + "'", myConnection);**

**myReader = myCommand.ExecuteReader(); while (myReader.Read())**

**{**

**col\_z = Convert.ToInt32(myReader["kol\_z"]); col\_l = Convert.ToInt32(myReader["kol\_l"]);**

**}**

**myReader.Close();**

**for (int j = 0; j < col\_z; j++)**

**{**

**dataGridView1.Columns.Add("z" + (j + 1), (j + 1).ToString()); dataGridView1.Columns["z" + (j + 1)].Width = 50;**

**}**

**for (int j = 0; j < col\_l; j++)**

**{**

**dataGridView1.Columns.Add("l" + (j + 1), "ЛР" + (j + 1)); dataGridView1.Columns["l" + (j + 1)].Width = 50;**

**}**

**myReader = null;**

**myCommand = new SqlCommand("select name\_discip, cod\_z, mark, name\_prepod, name**

**" +**

**"from eregister " +**

**"join discipline on name\_ds=discipline.cod join teachers on**

**name\_pr=teachers.cod join students on name\_st=students.nomer " +**

**"where name\_discip='" + tab + "'", myConnection); myReader = myCommand.ExecuteReader();**

**int i = 0; int k = 0;**

**bool fl = true;**

**while (myReader.Read())**

**{**

**int p = myReader.Depth; if (fl)**

**{**

**dataGridView1.Rows.Add(); dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString(); dataGridView1[0, i].Value = myReader["name"].ToString(); dataGridView1[1, i].Value = myReader["name\_discip"].ToString(); dataGridView1[2, i].Value = myReader["name\_prepod"].ToString(); k = 3;**

**fl = false;**

**}**

**if (k - 3 < col\_z)**

**if (myReader["mark"].ToString() == "1")**

**{**

**dataGridView1[k, i].Value = "+"; dataGridView1[k, i].Style.BackColor =**

**System.Drawing.Color.LightGreen;**

**}**

**else**

**if (myReader["mark"].ToString() == "0") dataGridView1[k, i].Value = "-";**

**if (k - 3 >= col\_z && k - 3 - col\_z < col\_l) if (myReader["mark"].ToString() != "n")**

**dataGridView1[k, i].Value = myReader["mark"].ToString();**

**k++;**

**if (k == 3 + col\_z + col\_l)**

**{**

**i++;**

**fl = true;**

**}**

**}**

**students\_count = i; myReader.Close();**

**}**

**catch (Exception exp)**

**{**

**label2.Text = exp.ToString();**

**}**

**}**

**Метод SqlDataReader.Read - Перемещает SqlDataReader к следующей записи. Метод SqlDataReader.Close - Закрывает объект SqlDataReader.**

**Формирование таблицы задолженностей по лабораторным работам:**

**private void FillTheTable\_Labs()**

**{**

**string tab = comboBox3.Text; int col = 0;**

**SqlDataReader myReader = null;**

**SqlCommand myCommand = new SqlCommand("select kol\_l from discipline where name\_discip='" + tab + "'", myConnection);**

**myReader = myCommand.ExecuteReader(); while (myReader.Read())**

**{**

**col = Convert.ToInt32(myReader["kol\_l"]);**

**}**

**myReader.Close();**

**string zadlist = string.Empty; ; for (int i = 0; i < col; i++)**

**zadlist += "'l" + (i + 1) + "',"; int index = zadlist.LastIndexOf(','); zadlist = zadlist.Remove(index);**

**bool fl = true; int k = 0;**

**try**

**{**

**myReader = null;**

**myCommand = new SqlCommand("select mark, name from eregister join discipline on name\_ds=discipline.cod join students on name\_st=students.nomer " +**

**"where name\_discip='" + tab + "' and cod\_z in (" + zadlist + ")",**

**myConnection);**

**myReader = myCommand.ExecuteReader(); int i = 0;**

**while (myReader.Read())**

**{**

**if (fl)**

**{**

**if (!dobavleno\_table4) dataGridView4.Rows.Add();**

**dataGridView4.Rows[i].HeaderCell.Value = (i + 1).ToString(); dataGridView4[0, i].Value = myReader["name"].ToString();**

**k = 0;**

**fl = false;**

**}**

**if (myReader["mark"].ToString() == "n") dataGridView4[k + 1, i].Value = false;**

**else**

**dataGridView4[k + 1, i].Value = true; k++;**

**if (k == col)**

**{**

**i++;**

**fl = true;**

**}**

**}**

**myReader.Close();**

**}**

**catch (Exception exp)**

**{**

**label2.Text = exp.ToString();**

**}**

**}**

**Учёт посещаемости:**

**string nomer1 = numericUpDown1.Value.ToString(); try**

**{**

**for (int i = 0; i < students\_count; i++)**

**{**

**SqlCommand myCommand = new SqlCommand("update eregister set mark=@z "**

**+ "where cod\_z='z"+nomer1+"' and name\_st="+(i+1)+"", myConnection); SqlParameter myParam1 = new SqlParameter("@z", SqlDbType.Char, 1);**

**if (Convert.ToBoolean(dataGridView2[0, i].Value)) myParam1.Value = '1';**

**else**

**myParam1.Value = '0';**

**myCommand.Parameters.Add(myParam1); myCommand.ExecuteNonQuery();**

**}**

**}**

**catch (Exception exp)**

**{**

**label4.Text = exp.ToString();**

**}**

**Подсчет количества пропусков:**

**if (!dobavleno\_pos)**

**{**

**dataGridView1.Columns.Add("Pos", "Пропуски"); dobavleno\_pos = true;**

**}**

**int count=0;**

**for (int i = 0; i < students\_count; i++)**

**{**

**for (int j = 3; j <= 3 + col\_l + col\_z; j++) if (dataGridView1[j, i].Value == "-")**

**count++;**

**dataGridView1["Pos", i].Value = count.ToString(); count = 0;**

**}**

**Определение полномочий преподавателя:**

**SqlDataReader myReader = null;**

**SqlCommand myCommand = new SqlCommand("select name\_prepod, polnomochiya from teachers", myConnection);**

**myReader = myCommand.ExecuteReader();**

**int i = 0;**

**while (myReader.Read())**

**{**

**comboBox1.Items.Add(myReader["name\_prepod"].ToString()); prepod\_mas[i] = myReader["polnomochiya"].ToString(); i++;**

**}**

**myReader.Close();**

**int n = comboBox1.SelectedIndex;**

**label6.Text = prepod\_mas[n];**

**Учёт лабораторных работ:**

**string tab = comboBox3.Text;**

**int nomer=comboBox2.SelectedIndex; try**

**{**

**SqlCommand myCommand;**

**SqlParameter myParam1 = new SqlParameter("@r1", SqlDbType.Char, 1); SqlParameter myParam4 = new SqlParameter("@nomer", SqlDbType.Int); myParam4.Value = nomer + 1;**

**for(int i=0;i<col\_l;i++)**

**if (dataGridView3[i, 0].Value != null)**

**{**

**myCommand = new SqlCommand("update eregister set mark=@r1 from eregister inner join discipline on name\_ds=discipline.cod "**

**+ "where name\_st=@nomer and name\_discip='" + tab + "' and cod\_z='l"+(i+1)+"'", myConnection);**

**myParam1.Value = dataGridView3[i, 0].Value; myCommand.Parameters.Add(myParam1); myCommand.Parameters.Add(myParam4); myCommand.ExecuteNonQuery(); myCommand.Parameters.Clear();**

**}**

**}**

**catch (Exception exp)**

**{**

**label4.Text = exp.ToString();**

**}**

**Добавление нового предмета:**

**string name\_ds = textBox1.Text;**

**int count = 0;**

**SqlDataReader myReader = null;**

**SqlCommand myCommand = new SqlCommand("select COUNT(\*) from discipline", Form1.myConnection);**

**myReader = myCommand.ExecuteReader(); while (myReader.Read())**

**{**

**count=Convert.ToInt32(myReader[""]);**

**}**

**myReader.Close();**

**string newcount = "0" + (count + 1);**

**myCommand = new SqlCommand("insert discipline (cod, name\_discip) values ('"+newcount+"', '"+name\_ds+"')", Form1.myConnection);**

**myCommand.ExecuteNonQuery(); label2.Text = "OK";**

**Создание журнала по предмету:**

**int col\_z = Convert.ToInt32(numericUpDown1.Value);**

**int col\_l = Convert.ToInt32(numericUpDown2.Value); SqlCommand myCommand;**

**for (int j = 0; j < 15; j++)**

**{**

**for (int i = 0; i < col\_z; i++)**

**{**

**myCommand = new SqlCommand("insert eregister values ('" + cod\_ds + "', 'z"**

**+ (i + 1) + "', 'n', '" + cod\_pr + "', " + (j + 1) + ")", Form1.myConnection); myCommand.ExecuteNonQuery();**

**}**

**for (int i = 0; i < col\_l; i++)**

**{**

**myCommand = new SqlCommand("insert eregister values ('" + cod\_ds + "', 'l"**

**+ (i + 1) + "', 'n', '" + cod\_pr + "', " + (j + 1) + ")", Form1.myConnection); myCommand.ExecuteNonQuery();**

**}**

**}**

**myCommand = new SqlCommand("update discipline set dob='1', kol\_z="+col\_z+", kol\_l="+col\_l+" where cod='"+cod\_ds+"'", Form1.myConnection);**

**myCommand.ExecuteNonQuery(); label5.Text = "OK";**