



МИНПРОСВЕЩЕНИЯ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тульский государственный педагогический университет им. Л.Н. Толстого»  
(ТГПУ им. Л.Н. Толстого)

Кафедра информатики и информационных технологий

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

**на тему:**

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ  
СИСТЕМЫ: «ЖУРНАЛ АКАДЕМИЧЕСКОЙ ГРУППЫ ВУЗА»**

Выполнил

студент группы 120871

очной формы обучения

факультета математики, физики и  
информатики

Куликов Константин Петрович

Тула — 2021

**Работа выполнена на факультете математики, физики и информатики  
ФГБОУ ВО «ТГПУ им. Л. Н. Толстого»**

Руководитель выпускной квалификационной работы — Ванькова  
Валентина Сергеевна, кандидат физико–математических наук, доцент  
кафедры информатики и информационных технологий

\_\_\_\_\_ В. С. Ванькова  
(Дата) (Подпись)

**Работа допущена к защите:**

Заведующий кафедрой информатики и информационных технологий

\_\_\_\_\_ Ю. И. Богатырева  
(Дата) (Подпись)

Защита состоится «\_\_\_\_\_» \_\_\_\_\_ 2020 года в учебном корпусе  
№ \_\_\_\_ ТГПУ им. Л. Н. Толстого, в аудитории № \_\_\_\_\_ в \_\_\_\_\_ часов.

Декан факультета математики, физики и информатики

\_\_\_\_\_ И. Ю. Реброва  
(Дата) (Подпись)

## Оглавление

ВВЕДЕНИЕ .....	4
ГЛАВА 1. МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	6
1.1 Общее описание этапов разработки информационной системы .....	6
1.2 Определение требований к информационной системе .....	7
1.3 Моделирование информационной системы .....	9
1.4 Средства и технологии разработки информационной системы .....	24
Выводы по главе 1 .....	25
ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЖУРНАЛ АКАДЕМИЧЕСКОЙ ГРУППЫ ВУЗА» .....	27
2.1 Выделение основных модулей и субмодулей .....	27
2.2 Архитектура информационной системы .....	29
2.3 Структура информационной системы .....	31
2.4 Реализация модулей информационной системы .....	33
2.5 Тестирование информационной системы .....	52
2.6 Развертывание информационной системы .....	53
2.7 Выводы по главе 2 .....	58
ЗАКЛЮЧЕНИЕ .....	60
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....	62
ПРИЛОЖЕНИЯ .....	65

## **ВВЕДЕНИЕ**

Сегодня в каждом ВУЗе России есть журнал академической группы. Его основная задача заключается в фиксации посещаемости занятий студентами, учета тем и описании общей структуры занятий.

Такой журнал существует исключительно в бумажном виде, что существенно ограничивает его возможности. С одной стороны, его легко заполнять, а с другой, его не очень удобно анализировать. Например, если необходимо получить информацию о посещаемости конкретного студента за семестр, то необходимо перелистать несколько страниц и при этом не запутаться в большом списке студентов. Если же преподаватель хочет узнать информацию о занятиях, которые он уже провел в разных группах, для составления отчетной ведомости ему необходимо взять журналы всех групп, в которых он ведет занятия и, соответственно, перенести информацию из этих журналов в ведомость.

Такой формат журнала академической группы не совсем удобен: слишком много действий нужно сделать, чтобы выбрать какие-то данные из него, да и легко ошибиться при выборке. Еще одним немаловажным фактором, играющим не в пользу бумажного журнала, является его надежность. Дело в том, что бумажный журнал легко повредить или потерять, что не очень хорошо, как для ВУЗа, так и для человека, ответственного за этот журнал, например, старосты группы.

Устранить все эти недостатки, связанные с удобством и безопасностью, можно с помощью переноса журнала академической группы в цифровой формат, т.е. можно создать информационную систему, способную интегрировать в себе описание академических групп, дисциплин, преподавателей, занятия, формы и методы их проведения. Помимо этого, информационная система должна быть максимально надежной. Разумеется, невозможно разработать информационную систему со 100% надежностью,

однако есть способы, позволяющие максимально приблизиться к хорошим показателям безопасности. Очевидно одно, электронная версия журнала академической группы должна быть на порядок надежнее бумажной.

Процесс разработки информационной системы не является тривиальной задачей и включает в себя как минимум пять основных этапов: сбор требований, проектирование, реализация, тестирование, эксплуатация и сопровождение. Для каждого из них далее по тексту работы дадим общее описание.

Объектом исследования данной работы является проблемная область ведения журнала академической группы ВУЗа.

Целью работы является моделирование и реализация информационной системы «Журнал академической группы ВУЗа».

Цель определяет следующие задачи:

1. Исследование предметной области «Журнал академической группы ВУЗа».
2. Разработка инфологической, логической и физической моделей данных, способных адекватно описать предметную область.
3. Выделение основных компонентов разрабатываемой информационной системы.
4. Поэтапная разработка каждого компонента информационной системы и реализация основных этапов.

Структура данной работы состоит из двух глав, списка источников и приложений. В первой главе исследуется предметная область и выделяются основные сущности будущей информационной системы, описывается модель данных и способ их реализации. Во второй главе описывается непосредственная реализация информационной системы на основании разработанной в первой главе модели, выделяются модули и субмодули, описывается архитектура и структура информационной системы.

# ГЛАВА 1. МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 1.1 Общее описание этапов разработки информационной системы

Гост 34.320-96 дает следующие определение информационной системы: «информационная система (INFORMATION SYSTEM): Концептуальная схема, информационная база и информационный процессор, составляющие вместе формальную систему для хранения и манипулирования информацией»[10].

Как было сказано ранее, есть пять основных этапов разработки информационной системы. Теперь стоит дать краткую характеристику каждому из них [11]:

1. **Определение требований к информационной системе:** это первый этап разработки и один из самых важных. На этом этапе определяются пользователи информационной системы и, в соответствии с их запросами, выделяется ключевой функционал под каждого пользователя и формируются требования к системе.
2. **Проектирование:** на этом этапе выполняется разработка модели данных, способной максимально полно и однозначно описать предметную область информационной системы. Если информационная система предусматривает наличие базы данных – она проектируется на этом этапе. При этом, все вопросы, связанные с функциональностью, способами хранения данных, видами баз данных, решаются на этапе определения требований к системе.
3. **Реализация:** на этом этапе выполняется непосредственная реализация информационной системы на основании той модели, что была получена при проектировании, на выбранном языке программирования. Язык программирования, СУБД и другие программные средства описываются в техническом задании, где также обосновывается их выбор.

- 4. Тестирование:** на этом этапе выполняется полное тестирование информационной системы. Целью данного этапа является определение корректности функционирования информационной системы в стандартных условиях (когда на вход подаются корректные данные), граничных условиях (когда на вход подаются редко используемые данные или какие-либо граничные значения) и критических условиях (когда входные данные не являются корректными). Создается модель тестирования, описывающая реакцию системы во время проведения тестирования.
- 5. Внедрение и сопровождение:** на этом этапе выполняется внедрение информационной системы в «экосистему» заказчика. Производится работа по обеспечению безопасности и надежности информационной системы. Проводится инструктаж персонала квалифицированными специалистами разработчика. Разработчик выполняет поддержку информационной системы в течение некоторого срока времени, установленного в договоре. Под поддержкой следует понимать выполнение следующих действий: установка и настройка информационной системы, выпуск и установка обновлений, предоставление инструкций по решению проблем или же самостоятельное решение проблем командой технической поддержки, администрирование и т. д. Вариантов может быть много, главное – все они должны быть прописаны в договоре.

## **1.2 Определение требований к информационной системе**

Нельзя разработать информационную систему, не имея представления о ее назначении, а что бы понять ее назначение, необходимо разобраться с тем, кто будет использовать эту систему.

Мы моделируем ведение журнала академической группы и знаем, что его заполнением занимаются, как правило, студенты (старосты групп) и

преподаватели. Помимо этого, журнал могут просматривать представители деканата, например, с целью контроля посещаемости или добавления каких-либо записей (новых студентов, преподавателей, дисциплин и т. д.).

Определившись с пользователями информационной системы, можно смело говорить о какой-либо ее функциональности. Начнем с самого верхнего уровня привилегий и, постепенно будем спускаться вниз по уровням. На данном этапе разработки будем говорить о следующих объектах (сущностях разрабатываемой информационной системы): представители деканата, преподаватели.

Набольшими привилегиями обладают представители деканата (их, в дальнейшем, мы будем называть операторами) и для них предполагается следующая функциональность:

1. Управление списком студентов и преподавателей;
2. Управление списком академических групп;
3. Управление списком дисциплин;
4. Назначение дисциплин академическим группам;
5. Назначение дисциплин преподавателям;
6. Распределение студентов по подгруппам;
7. Распределение учебной нагрузки по преподавателям;
8. Просмотр отчета о посещаемости и о занятиях;

Следующими по уровню привилегий, идут преподаватели:

1. Запись, редактирование и удаление информации о занятиях;
2. Проверка и заполнение таблицы посещаемости занятий;
3. Выгрузка отчетов по посещаемости;
4. Просмотр и выгрузка информации о проведенных занятиях в отчетную ведомость.



И последним видом пользователей системы будет - студент. Здесь нужно обратить внимание, что студентов мы делим на старосту и все остальных. Староста обладает возможностью:

1. Добавление новых студентов;
2. Занесение информации о посещаемости занятий;
3. Получение общих сведений о занятиях.

Все остальные студенты могут просматривать информацию о занятиях и накопленных баллах.

В рамках данной выпускной квалификационной работы будет выполнена реализация функционала только для первых двух видов пользователей, однако, разработанная в процессе моделирования схема данных в дальнейшем позволит реализовать функционал для студентов.

### **1.3 Моделирование информационной системы**

#### **Инфологическая модель данных**

После прохождения этапа определения требований, на котором были сформированы основные требования к функционалу информационной системы, переходим к этапу создания инфологической модели.

На этом этапе нам предстоит разработать описание схемы данных нашей информационной системы в максимально понятном и однозначном для понимания виде. Стоит отметить, что инфологическая модель является абсолютно не зависимой от физической реализации схемы данных.

Для представления схемы данных будем использовать EER-диаграмму, которая позволит нам сделать концептуальное описание предметной области.

Сначала стоит определить основные понятия, которыми мы будем оперировать в процессе построения инфологической модели [1]:

- **Сущность** – есть объект базы данных, используемый для хранения каких-либо данных информационной системы. Для упрощения понимания, можно сказать, что сущность — это таблица в реляционной базе данных. Для нашей информационной системы примерами сущностей могут быть студенты, преподаватели, академические группы, дисциплины, сведения о посещаемости и т.д. Сущность описывается совокупностью атрибутов.
- **Атрибут сущности** – это свойство, которое описывает сущность. Атрибут может быть представлен числом, символом, текстом или датой. Важно, что все данные, хранящиеся в атрибуте, обладают одним типом и одними свойствами. В физической схеме базы данных атрибут называют колонкой.

Для каждой сущности должен быть задан адекватный перечень атрибутов, для того чтобы максимально однозначно и детально охарактеризовать объект исследования.

Мы уже приводили примеры сущностей для нашей информационной системы, теперь стоит описать примеры атрибутов. Для студента атрибутами могут быть: фамилия, имя, отчество, дата рождения, пол, академическая группа и т. д. Атрибутов может быть множество, главное, чтобы они четко описывали объект в рамках заданной функциональности информационной системы. Например, добавление к студенту атрибута знак зодиака не является адекватным описанием объекта. Такой атрибут избыточен, так как нам, с учетом поставленной функциональности, абсолютно не важно кто студент по гороскопу.

Так же стоит выделить основные виды атрибутов [2]:

- **Первичный ключ** - атрибут или набор атрибутов, однозначно идентифицирующий запись в таблице (сущности). Первичный ключ сможет быть как простым, т. е. состоящим из одного атрибута, так и

составным, т. е. состоящим из нескольких атрибутов. Важно, чтобы первичный ключ был максимально уникальным для конкретного объекта сущности. Например, для студента первичным ключом может быть номер зачетной книжки. Если говорить о составном ключе, то здесь стоит привести в пример сущность описывающая посещаемость занятий. В этой сущности используется идентификатор студента и идентификатор занятия. Такая запись позволит гарантировать, что информация о студенте и посещенном им занятии не будет записана дважды в таблицу.

- **Внешний ключ:** внешним ключом называется атрибут, ссылающийся на первичные ключи другой сущности. Например, для таблицы посещаемости занятий, внешними ключами будут являться ее первичные ключи, т. е. идентификатор студента и идентификатор дисциплины.
- **Связь** – описание формы взаимодействия двух и более сущностей. Связи делятся на следующие виды:
- **Один к одному:** такой вид связи говорит о том, что объекту из сущности А соответствует только один объект из сущности Б. Например, в нашей информационной системе будет отдельная сущность для описания информации о данных для авторизации студентов и преподавателей. Назовем ее Users. Ее полями будут некоторый идентификатор, логин, email, пароль и роль пользователя в системе. Сущность Users связывается по типу 1:1 с сущностями StudentProfile (информация о студенте) и TeacherProfile (информация о преподавателе). При этом, важно отметить, что такая связь является обязательной. То есть, сущность с информацией о студенте не может содержать запись с идентификатором, которого нет в сущности Users. Точно также и в TeacherProfile.

- **Один ко многим:** такой вид связи говорит о том, что объекту из сущности А может соответствовать множество объектов из сущности Б. Например, один студент может быть только в одной академической группе и академическая группа может содержать множество студентов.
- **Многие ко многим:** при таком виде связи множеству объектов из сущности А может соответствовать множество объектов из сущности Б. Как правило, такой вид связи реализуется с помощью таблицы посредника, назовем ее С. Тогда Таблица А относится к С как один ко многим, и таблица Б также относится к С, как один ко многим. В результате, в С получается сопоставление множества объектов А с множеством объектов Б.

Теперь можно приступить непосредственно к разработке модели информационной системы.

Исходя из представленных выше требований к функционалу, видно, что выделяются три большие группы: «Операторы», «Студенты», «Преподаватели». Эти группы должны быть разделены между собой привилегиями. Следовательно, нам необходимо ввести ролевую политику с тремя уровнями привилегий, соответствующих группам.

Для реализации ролевой политики, помимо таблицы пользователей введём таблицу ролей (Рисунок 1).

Для обеспечения закрытости ИС нужно исключить ситуацию, когда любой человек, не связанный с ВУЗом, смог бы войти в нее или зарегистрироваться. Для этого реализуем следующие шаги:

1. Оператор информационной системы сам добавляет новых пользователей.
2. Аккаунт вновь добавленного пользователя не является активным, т.е. никто не сможет войти, используя данные пользователя.

3. После добавления каждому новому пользователю на почту отправляется письмо с уведомлением о регистрации в ИС, и ему предоставляется ссылка для активации аккаунта.
4. После перехода по ссылке пользователь должен придумать себе пароль.

Для реализации такого функционала нам потребуется таблица *AccountActivation* (Рисунок 1), которая будет содержать ID новых пользователей и хэш секретного кода, который был сгенерирован при добавлении нового пользователя и отправлен также на email.

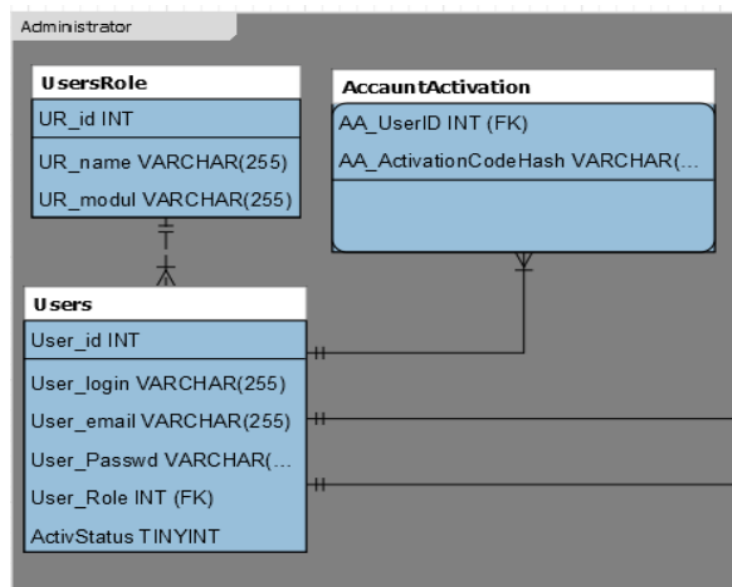


Рисунок 1 - Слой “Administrator” в схеме данных ИС

В результате мы получаем административный слой информационной системы, прямо отвечающий за авторизацию пользователей и регистрацию новых. Таблицы этого слоя реализуют возможность разделения прав доступа на уровне модели ИС.

Следующим важным слоем ИС является слой *Users* (Рисунок 2). Он описывает таблицы с информацией о студентах и преподавателях. Отметим, что учетной записи оператора соответствует запись в таблице *TeacherProfile*.

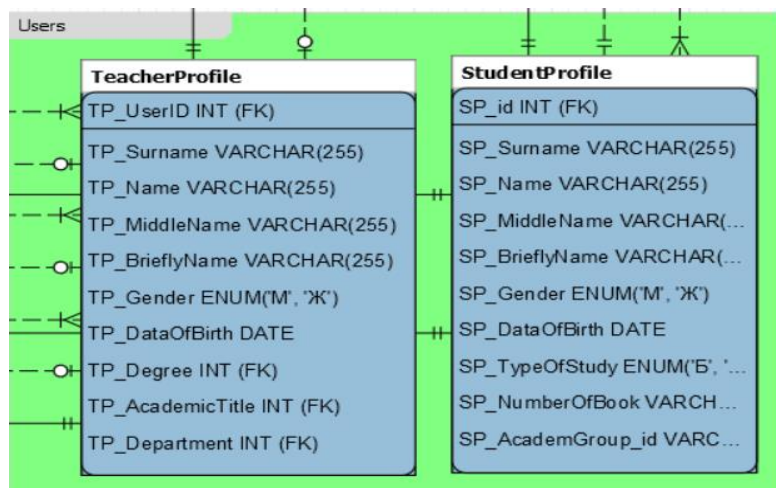


Рисунок 2 - Слой “Users”

Таблицы слоя *Users* призваны описать каждый профиль пользователя ИС так, чтобы это удовлетворяло требованиям ИС и удобству использования.

Теперь рассмотрим слой *Reference* (Рисунок 3). На этом слое расположены вспомогательные таблицы (словари) ИС. Их задача «разгрузить» основные таблицы информационной системы и обеспечить возможность нормализации модели данных до формы Бойса-Кодда.

На этом слое расположены следующие таблицы:

1. **LessonNumber** – таблица (словарь), описывающая расписание пар. Каждой паре соответствует ее время ее начала и конца.
2. **AcademicTitle** – словарь должностей для преподавателей.
3. **AcademicDegree** – словарь ученых степеней для преподавателей.
4. **Facultets** – таблица факультетов с указанием аббревиатуры и декана.
5. **Departaments** – таблица с описанием кафедр. Каждая кафедра имеет аббревиатуру, факультет и заведующего.
6. **SpecialtyList** – таблица специальностей.
7. **SpeciltyProfile** – таблица направлений (профилей подготовки).
8. **TypeLesson** – словарь с описанием типов занятий. Например, лекции, практические занятия и т.д.

9. **FormOfControl** – Словарь с описанием форм контроля для дисциплин. Например, зачеты, экзамены и т.д.
10. **FormatOfStudy** – Словарь с формами обучения: очная, заочная, очно-заочная.
11. **Discipline** – таблица дисциплин. Описывает дисциплины названием, ведущим преподавателем, и кратиким описанием преподаваемого предмета.

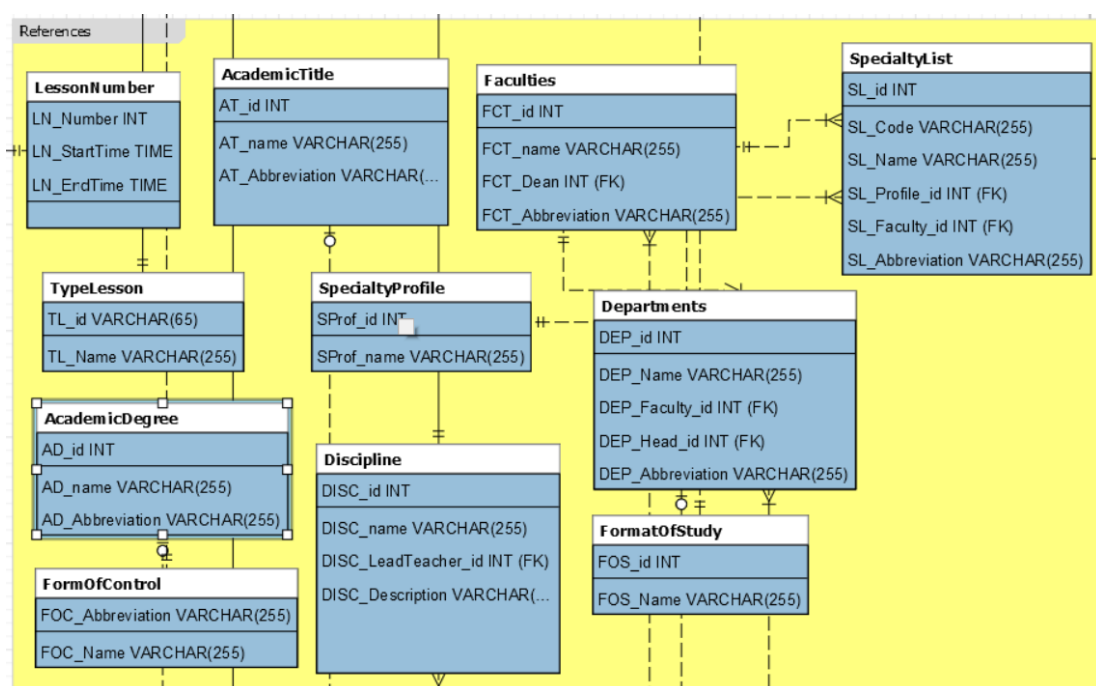


Рисунок 3 - Слой “Reference”

Следующим слоем модели ИС является *Academic Groups* (Рисунок 4). Этот слой описывает академические группы и наиболее важные сущности для академических групп. Слой оперирует четырьмя таблицами, которые следует рассмотреть подробнее, как каждая из них является существенно важной частью ИС:

1. **AcademicGroups** – сущность описывает академическую группу ВУЗа следующими свойствами: код группы (*AG\_Code*), специальность (*AG\_specialty*), год начала и окончания подготовки (*AG\_YearOfStart* и *AG\_YearOfEnd*), текущий курс (*AG\_NumCourse*), куратор

(*AG\_Curator*), староста группы (*AG\_Headman*), форма обучения (*AG\_FormOfStudy*).

2. **StudyLoad** – таблица с описанием учебной нагрузки группы. Учебная нагрузка обладает следующими свойствами: дисциплина, вид занятий, количество часов, преподаватель, код группы, сведения о дополнительной нагрузке, идентификатор нагрузки, номер семестра, форма контроля.
3. **SubGroups** – таблица, отвечающая за формирование подгрупп. Для каждой учебной нагрузки формируется своя подгруппа.
4. **StudentListInSubgroups** – таблица распределения студентов по подгруппам.

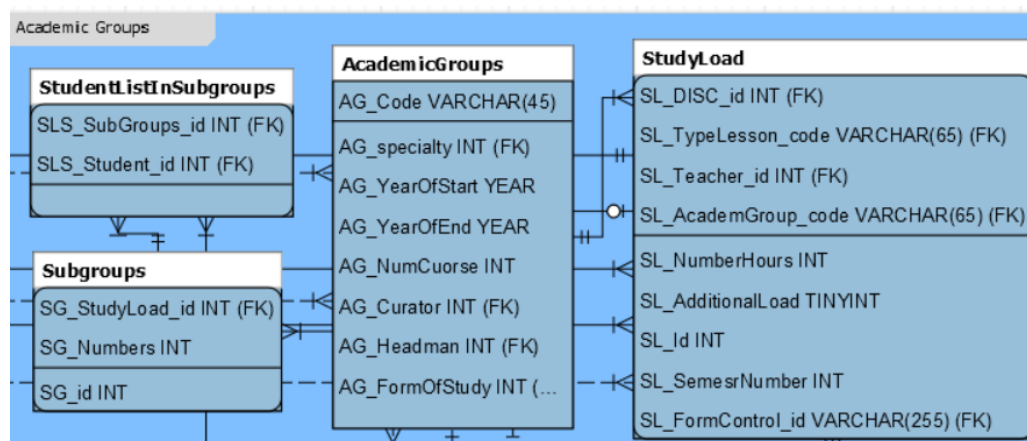


Рисунок 4 - Слой “Academic Groups”

Завершающим слоем модели информационной системы является слой *Journal* (Рисунок 5). На этом слое представлена информация о занятиях и посещаемости студентов.

Таблица *LessonInfo* содержит следующие данные: идентификатор занятия, дата проведения занятия, идентификатор учебной нагрузки, тема занятия, номер пары и время начала и окончания индивидуальных занятий.



Таблица *AttendanceLesson* содержит: идентификатор занятия, идентификатор студента, количество часов, которые посетил студент, полученные студентом в ходе занятия баллы и комментарий преподавателя.

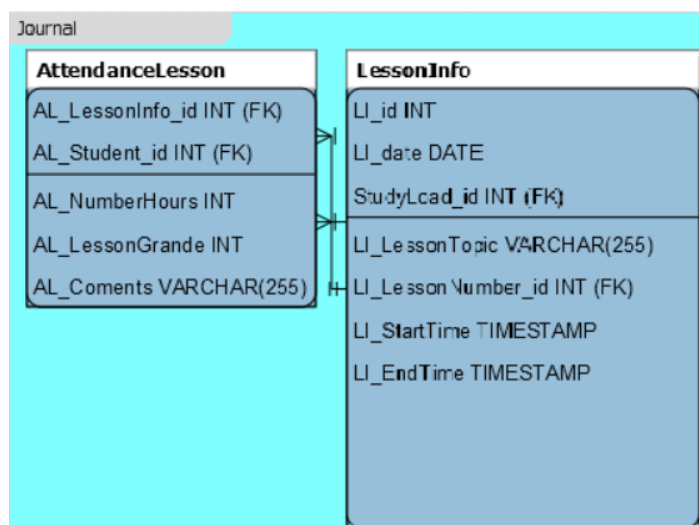


Рисунок 5 - Слой “Journal”

Как можно увидеть выше, таблиц получилось не мало. Однако это необходимым минимум для описания информационной системы, способной удовлетворять поставленным условиям.

Так же стоит отметить, что есть возможность расширения модели данной информационной системы, следствием чего станет усиление ее функционала. Например, можно добавить возможность вывода информации о занятиях с использованием расписаний самих занятий или ввести функцию создания отчета про БРС (основы мы для этого заложили, поскольку в таблице посещаемости ведется учет баллов). Задач может быть много, и самым важным является выделение границ моделирования. Как известно, выделяют два компонента границ:

- Ширина детализации – определяет внешние границы моделируемой предметной области (в нашем случае этот тот самый примерный функционал, которым можно расширить информационную систему).

- Глубина детализации – определяет степень подробности, с которой можно выполнять декомпозицию задачи.

Таким образом, в результате описания предметной области с целью создания инфологической модели, выделены 22 сущности (Таблица 1), и их совокупность способна в полной мере описать разрабатываемую информационную систему.

Таблица 1 - Список сущностей

<u>Таблица</u>	<u>Комментарий</u>
<u>AcademicDegree</u>	Список ученых степеней
<u>AcademicGroups</u>	Академические группы
<u>AcademicTitle</u>	Должности/Ученые степени
<u>AccauntActivation</u>	Таблица активации аккаунта
<u>AttendanceLesson</u>	Посещаемость
<u>Departments</u>	Кафедры
<u>Discipline</u>	Таблица дисциплин
<u>Faculties</u>	Факультеты
<u>FormatOfStudy</u>	Форма обучения
<u>FormOfControl</u>	Таблица форм контроля
<u>LessonInfo</u>	Информация о занятии
<u>LessonNumber</u>	Расписание пар
<u>SpecialtyList</u>	Список специальностей
<u>SpecialtyProfile</u>	Таблица профилей(направленностей)
<u>StudentListInSubgroups</u>	Списки студентов по подгруппам
<u>StudentProfile</u>	Студенты
<u>StudyLoad</u>	Учебная нагрузка дисциплины
<u>Subgroups</u>	Подгруппы
<u>TeacherProfile</u>	Профиль преподавателя
<u>TypeLesson</u>	типы занятий
<u>Users</u>	Пользователи
<u>UsersRole</u>	Таблица ролей пользователей

На следующем этапе нам предстоит разработать логическую и физическую модели данных. Там же опишем сущности с помощью атрибутов.

Отметим, что в дальнейшем не будем рассматривать роль «Студенты», так как особой функциональной нагрузки она не несет.

## Логическая и физическая модели данных

На этом этапе нам предстоит решить следующие задачи: описание сущностей с помощью атрибутов, разработка логической и физической моделей данных.

Если с описанием сущностей все понятно, то для описания логической и физической модели следует дать подробное разъяснение.

«Логическая модель представляет собой графическое описание схемы данных с указанием всех атрибутов и связей.» [3]

«Физическая модель представляет собой ту же самую логическую модель данных, выраженную в терминах описания СУБД»[6]. Иными словами, физическая модель может быть представлена, как схема данных с указанием технических деталей реализации или же, как, например, SQL – скрипт, позволяющий импортировать базу данных на сервер.

Теперь опишем все сущности и их атрибуты. Для описания модели данных можно воспользоваться инструментом MySQL Workbench. Он позволяет создать наглядную схему данных, а также выгрузить эту схему на сервер с сохранением всех сущностей, атрибутов и связей.

Сначала опишем все сущности, а затем построим EER – схему данных.

Таблица 2 содержит описание сущности «Список ученых степеней» в виде совокупности трех атрибутов. Первый служит уникальным идентификатором для записи и его типом является число. Вторым (AD\_name) является полным названием ученой степени, например, кандидат физико-математических наук, и является строкой в 255 символов. Третий (AD\_BrieflyName) является кратким обозначением, например, канд. физ.-мат. наук, и так же представлен строкой в 255 символов. Ни один из атрибутов не может быть NULL – т.е. не определенным. Из этой сущности будет браться информация об ученой степени преподавателя для описания сущности «Преподаватель»

Таблица 2 - Сущность «Список ученых степеней»

Столбец	Тип	Null	Комментарии
AD_id (Первичный)	int(11)	Нет	ИД степени
AD_name	varchar(255)	Нет	Ученая степень
AD_BrieflyName	varchar(255)	Нет	Краткое обозначение

Таблица 3 описывает сущность «Академические группы». Это базовая сущность предназначена для описания студента в рамках нашей информационной системы. AG\_Code описывает код академической группы и является первичным ключом для данной сущности. Атрибуты AG\_specialty, AG\_Curator, AG\_Headman и AG\_FormOsStudy являются ссылочными и содержат в себе ссылки на другие сущности, которые будут описаны в дальнейшем.

Таблица 3 - Академические группы

Столбец	Тип	Null	По умолчанию	Ссылки на	Описание
AG_Code (Первичный)	varchar(45)	Нет			Код группы
AG_specialty	int(11)	Нет		specialtylist -> SL_id	Специальность
AG_YearOfStart	year(4)	Нет			Год начала
AG_YearOfEnd	year(4)	Нет			Год окончания
AG_NumCuorse	int(10)	Нет	1		Курс
AG_Curator	int(11)	Да	NULL	teacherprofile -> TP_UserID	Куратор
AG_Headman	int(11)	Да	NULL	studentprofile -> SP_id	Староста
AG_FormOfStudy	int(11)	Нет		formatofstudy -> FOS_id	Форма обучения

Четвертая таблица описывает сущность «Ученое звание». Эта сущность нужна для описания преподавателей.

Таблица 4 - Сущность «Учёное звание»

Столбец	Тип	Null	Комментарии
AT_id (Первичный)	int(11)	Нет	
AT_name	varchar(255)	Нет	Ученое звание

AT_BrieflyName	varchar(255)	Нет	Краткое обозначение
----------------	--------------	-----	---------------------

Таблица 5 описывает информацию о посещаемости занятий студентами. Когда мы говорили о составных ключах, то рассматривали эту таблицу. Здесь ключ представлен парой: ИД занятия и ИД студента. Иными словами, в эту таблицу могут попасть только уникальные комбинации этих двух полей. Действительно, мы не можем отметить 2 раза студента на одной паре. Так же здесь приведены дополнительные свойства, такие как количество академических часов, проведенных студентом на паре, заработанные им баллы и комментарий от преподавателя.

Таблица 5 - Учет посещаемости

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
AL_LessonInfo_id (Первичный)	int(11)	Нет		lessoninfo -> LI_id	ИД занятия
AL_Student_id (Первичный)	int(11)	Нет		studentprofile -> SP_id	ИД Студент
AL_NumberHours	int(10)	Да	0		Количество астрономических, которые посетил студент
AL_LessonGrande	int(11)	Да	0		Заработанные баллы
AL_Coments	varchar(255)	Да	NULL		Комментарий преподавателя

Таблица 6 описывает сущность «Кафедры» и содержит информацию о кафедрах и используется для описания преподавателя. Обратите внимание, что заведующий кафедрой может быть не указан. Это сделано для того, чтобы в момент добавления новой кафедры существовала возможность сначала назначить на нее преподавателей, а уже потом выбрать из них заведующего.

Таблица 6 - Сущность «Кафедры»

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
DEP_id (Первичный)	int(11)	Нет			

DEP_Name	varchar(255)	Нет			Название кафедры
DEP_Faculty_id	int(11)	Нет		faculties -> FCT_id	Факультет
DEP_Head_id	int(11)	Да	NULL	teacherprofile -> TP_UserID	Заведующий кафедры
DEP_Abbreviation	varchar(255)	Нет			Аббревиатура

В таблице 7 описывается сущность «Информация о занятии». Она служит для заполнения сведений о проведенных занятиях.

Таблица 7 - Сущность «Информация о занятии»

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
LI_id ( <i>Первичный</i> )	int(11)	Нет			ид занятия
LI_Discipline_id	int(11)	Нет		discipline -> DISC_id	Дисциплина
LI_Teacher	int(11)	Нет		teacherprofile -> TP_UserID	Преподаватель
LI_date	date	Нет			Дата проведения занятия
LI_AcademGroup_code	varchar(45)	Нет		academicgroups -> AG_Code	Академическая Группа
LI_TypeLesson_id	varchar(65)	Нет		typelesson -> TL_id	Тип занятия
LI_LessonTopic	varchar(255)	Нет			Тема занятия
LI_LessonNumber_id	int(10)	Да	NULL	lessonnumber -> LN_Number	Номер пары
LI_StartTime	timestamp	Да	NULL		Время начала занятия
LI_EndTime	timestamp	Да	NULL		Время окончания занятия

В таблице 8 описывается сущность «Учебная нагрузка». Она призвана описать учебную нагрузку преподавателя в конкретной группе и по конкретному виду занятий. Мы видим, что в данной сущности присутствует составной ключ из четырех элементов: ИД дисциплины (из таблицы дисциплин), вид занятий (из таблицы видов занятий), преподаватель (из таблицы преподавателей), код академической группы (из таблицы

академических групп). Такое обильное количество элементов составного ключа обусловлено тем, что преподаватель может вести в одной группе более одного вида занятий, например, лекции и лабораторные работы по разным предметам.

Таблица 8 - Сущность «Учебная нагрузка»

Столбец	Тип	Null	Ссылки на	Комментарии
SL_DISC_id ( <i>Первичный</i> )	int(11)	Нет	discipline -> DISC_id	id дисциплины
SL_TypeLesson_code ( <i>Первичный</i> )	varchar (65)	Нет	typelesson -> TL_id	Вид занятия
SL_NumberHours	int(10)	Нет		Количество часов
SL_Teacher_id ( <i>Первичный</i> )	int(11)	Нет	teacherprofile -> TP_UserID	Преподаватель
SL_AcademGroup_code ( <i>Первичный</i> )	varchar (65)	Нет	academicgroups -> AG_Code	Академическая группа
SL_AdditionalLoad	tinyint(4)	Да		Дополнительная нагрузка

Итак, выше приведены описания 7 из 21 таблицы. Описывать все их здесь нецелесообразно. Все они будут приведены в приложении.

Теперь стоит разобраться зачем же нам такое количество таблиц. Ведь с первого взгляда, например, информацию об ученой степени и звании для преподавателя, можно хранить в одной таблице вместе с данными о преподавателе. Да, так можно хранить данные. Однако, в таком случае нарушится нормализация базы данных. Нормализация – это процесс удаления избыточных данных и приведение базы данных к минимальной избыточности. «Под избыточностью понимается ситуация, когда в базе данных хранятся повторяющиеся данные в нескольких местах.» [3]

В идеале база данных хорошей информационной системы должна находиться в третьей нормальной форме. Поэтому на этапе проектирования сущностей, мы старались избавиться от дублирования атрибутов и создавали дополнительные таблицы. Такой подход придает определенную гибкость для работы с данными. Однако избыточная нормализация может усложнить работу с базой.

В результате оптимизации на основании построенной нами модели получаем EER – диаграмму (Рисунок 1), описывающую схему данных информационной системы. Для упрощения графического предоставления были исключены поля, не относящиеся к первичным и вторичным ключам.

Полная схема данных будет приведена в приложении к данной работе.

#### 1.4 Средства и технологии разработки информационной системы

Для удобного проектирования информационной системы и разработки функции ввода данных воспользуемся следующим программным обеспечением:

1. **MySQL Workbench:** инструмент для проектирования баз данных. Позволяет выполнять визуальное проектирование схемы данных с последующей загрузкой созданной модели на сервер баз данных. Распространяется под свободной лицензией GNU/GPL, что позволяет использовать данный инструмент на безвозмездной основе.
2. **Open Server:** свободно распространяемая серверная платформа, включающая в себя ряд программных средств, предназначенных для развёртывания серверной архитектуры для разработки, отладки и тестирования веб-проектов. Конкретно для нашего проекта на текущем этапе будем использовать следующие программные средства из набора Open Server:
  - **Nginx 1.14** – веб – сервер. Распространяется под BSD – подобной лицензией из двух пунктов.
  - **PHP 7.1 (x64)** – скриптовый язык программирования общего назначения. Распространяется свободно, под собственной лицензий PHP.
  - **MariaDB 10.3** – система управления базами данных (СУБД). Является ответвлением СУБД MySQL и почти полностью с ней совместима.



Является свободным программным обеспечением, распространяющемся под лицензией GNU GPL и GNU LGPL.

3. Для программной реализации функций ввода данных можно воспользоваться любым свободным редактором кода с поддержкой HML, CSS, PHP и JavaScript. В данном же проекте использовался **JetBrains PHPStorm**. Данный редактор не является свободным и бесплатным. Однако, компания дает возможность получить бесплатную лицензию для учащихся и преподавателей из аккредитованных учебных заведений.

Помимо инструментов разработки нам понадобятся фреймворки и плагины для программной реализации некоторых компонентов:

1. **Bootstrap 4** – набор инструментов с открытым исходным кодом, позволяющий создавать адаптивные веб-сайты;
2. **Google reCAPTCHA v2** – бесплатная система Google, позволяющая защитить сайт от спама и DDoS атак;
3. **jQuery (версия 1.11.3)** – набор JavaScript функций, позволяющий упростить разработки сайта. В нашем случае преимущественно будут использоваться асинхронные JavaScript – запросы (AJAX);
4. **DataTable** (версия 1.10.24) – плагин jQuery, позволяющий создавать удобные и многофункциональные таблицы с данными.
5. **Select2** (версия 4.1) – позволяет создавать стилизованные select – элементы для удобного использования пользователем системы.
6. **Библиотека PhpSpreadsheet** – позволяет работать с XLS/XLSX файлами, ее мы будем использовать для выгрузки месячной ведомости.

## Выводы по главе 1

1. Разработка информационной системы представляет собой сложный и многоступенчатый процесс. Каждый следующий этап зависит от качественного выполнения предыдущего. Например, нельзя создать полноценную модель данных, не имея представления о функционале, закладываемом в эту модель.
2. Описание функционала информационной системы должно выполняться по ГОСТу и опираться на детальный анализ предметной области.
3. После описания функционала информационной системы следует осуществлять инфологическое, логическое и физическое проектирование данных. Построенная модель позволит в дальнейшем, с помощью средств программирования, создать функциональную информационную систему.
4. Средства и инструменты программирования для разработки информационной системы подбираются в зависимости от ее назначения, функционала и перспектив использования.

## ГЛАВА 2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «ЖУРНАЛ АКАДЕМИЧЕСКОЙ ГРУППЫ ВУЗА»

### 2.1 Выделение основных модулей и субмодулей

Информационная система «Журнал академической группы ВУЗа» предполагает наличие двух основных модулей системы: «Оператор», «Преподаватель» (Рисунок 6). Есть еще и третий модуль – «Студенты», но на этапе моделирования мы от него отказались, ввиду отсутствия функциональной нагрузки. Вопросы его реализации не являются критичными и могут быть сняты в любой момент без дополнительных изменений в структуре базы данных.

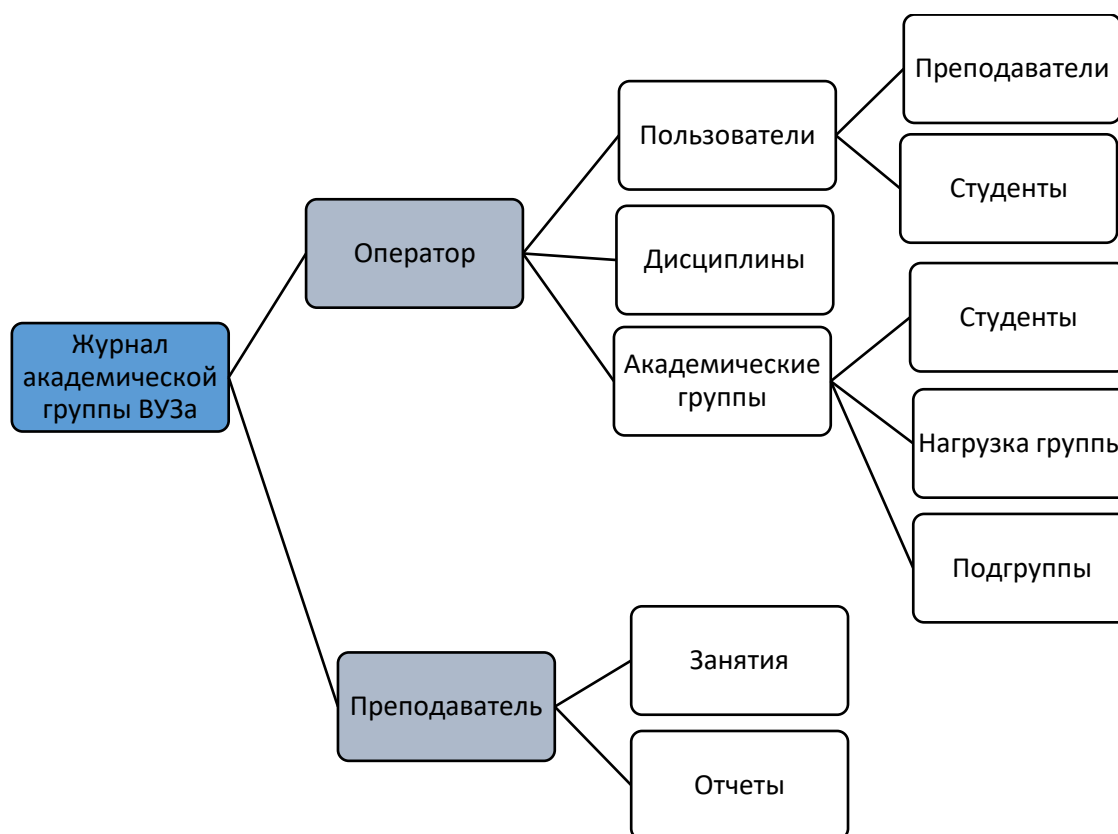


Рисунок 6 - Основные модули и субмодули системы

Модуль «**Оператор**» является базовым для информационной системы, так как отвечает за обслуживание всей системы. Данный модуль соответствует

роли «оператора» и призван обеспечить заполнение информационной системы данными, необходимыми для ее корректной работы.

Как видно из рисунка 6, модуль «Оператор» делится на 3 субмодуля: «Пользователи», «Дисциплины» и «Академические группы». Рассмотрим каждый из них подробнее.

Субмодуль «**Пользователи**» отвечает за добавление информации о пользователях в систему. Как видно из рисунка 6, этот субмодуль в свою очередь делится на субмодули «Студенты» и «Преподаватели». Каждый из них отвечает за добавление, редактирование и удаления, советуемого названию модули пользователя.

Субмодуль «**Дисциплины**» отвечает за управление (добавление, редактирование и удаление) списком дисциплин.

Субмодуль «**Академические группы**» один из самых сложных и важных субмодулей системы, отвечающий за управление академическими группами. Данный субмодуль делится на:

1. «Студенты» – субмодуль управления списком студентов академической группы.
2. «Нагрузка» – субмодуль распределения разгрузки дисциплин академической группы между преподавателями.
3. «Подгруппы» – субмодуль создания подгрупп на основе данных субмодуля «Нагрузка» и распределения студентов академической группы между созданными подгруппами.

Помимо всего, субмодуль «Академические группы» позволяет экспортировать данные об академических группах в Excel, PDF и на печать.

Модуль «**Преподаватель**» обеспечивает функциональность, позволяющую реализовать требования, описанные в разделе 1.2 главы 1 для преподавателей. Он делится на два важных субмодуля:

1. «**Занятия**» - субмодуль реализующий управление информацией (добавление, редактирование и удаление) о приведённых преподавателем занятиях, а также заполнением сведений о посещаемости занятий и накопленных студентами баллах.
2. «**Отчеты**» - данный субмодуль реализует функционал выгрузки ответов о посещаемости, баллах, проведенных занятиях.

## **2.2 Архитектура информационной системы**

Определившись с модулями информационной системы, можно переходить к описанию ее архитектуры. Архитектура любой информационной системы призвана описать модель, структуру, взаимосвязь компонентов и выполняемые ими функции.

В предыдущем разделе мы рассмотрели описание компонентов (модулей), но не затрагивали вопрос их внутренней реализации, а также способов взаимодействия.

Стоит отметить, что основные пути взаимодействия модулей заложены на уровне схемы базы данных и представляются вводе связей сущностей. Так же есть способы программного взаимодействия, о которых пойдет речь дальше.

По архитектуре данная информационная система относится к клиент-серверным приложениям. Это означает, что ее разработка разбивается на два этапа [16]:

1. Клиентская часть (Front-end) – отвечает за взаимодействие системы с пользователем: отрисовка графического интерфейса, отображение данных, формирование запросов к серверу.
2. Серверная часть (Back-end) – отвечает за взаимодействие системы с базой данных: добавление, удаление, редактирование, отображение

данных, валидация запросов и, в некоторых случаях, формирование интерфейса.

Для корректной работы информационной системы серверная и клиентская часть должны быть «связаны». Что бы реализовать такую связь воспользуемся подходом AJAX (Asynchronous Javascript and XML), который заключается в том, что серверная часть представляет из себя набор так называемых “Endpoint-ов” или, как мы их будем называть, API (application programming interface), к которым клиентская часть может посылать запросы и получать в ответ нужную информацию.

На рисунке 7 изображена схема взаимодействия серверной и клиентской части информационной системы. Как видно из рисунка, AJAX поселяет на сервер запросы вида GET или POST, сервер обрабатывает полученные параметры формирует и отправляет запрос к базе данных, база данных в свою очередь возвращает ответ серверу, после чего тот формирует данные ответа в зависимости от задачи (html или json) и возвращает данные AJAX компоненту, который и выполнял запрос. AJAX компонент, при необходимости преобразует данные и показывает их пользователю.

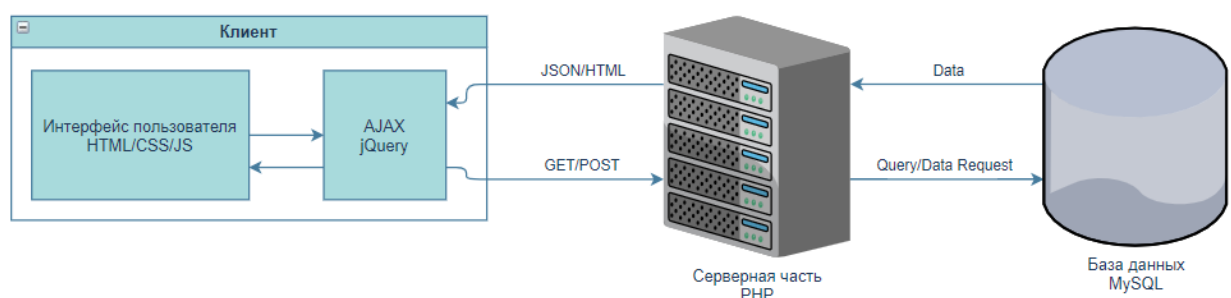


Рисунок 7 - Модель взаимодействия «Клиент-сервер»

Здесь стоит ввести следующую договоренность: GET запросы мы будем использовать только для получения данных (например, список студентов), т. к. GET-запросы не позволяют посылать большой объем данных и не обладают достаточной надежностью; POST-запросы будут использоваться для добавления и обновления информации.

Ответ сервера преимущественно будет в формате json, так как это наиболее удобный формат обработки массива данных. И, к тому же, на клиентской части для отрисовки таблиц будет задействована библиотека DataTables, которая умеет получать данные через AJAX – запросы, однако требует, чтобы все одни были в формате JSON.

В некоторых сложных ситуациях потребуется, чтобы сервер вернул html фрагмент интерфейса с уже подставленными данными. Это обусловлено тем, что иногда придется строить сложные представления, и их формирование на клиентской части не является целесообразным. Например, таблица посещаемости студентов является достаточно сложной для формирования, так как помимо динамически меняемых строк, есть еще и динамически изменяемые столбцы, которые нужно согласовывать со строками, а также придется использовать несколько запросов к базе данных для ее отображения, поэтому более удобным будет сгенерировать таблицу на сервере, а затем ставить ее в нужное место в html документе.

Серверная часть информационной системы представляет из себя набор из API, которые направлены на выполнение той или иной задачи. У каждого модуля свой набор API, что обусловлено различием в функциональности.

## **2.3 Структура информационной системы**

Информационная система «Журнал академической группы ВУЗа» является закрытой. Это означает, что самостоятельная регистрация пользователей в системе не допускается, как и не допускается наличие ролевой политики «Гость». Такой подход обусловлен обеспечением безопасности информационной системы.

Как было сказано в первом разделе второй главы, система разделена на модули. Эти модули помимо реализации функционала разграничиваются между собой ролевой политикой.

При создании нового пользователя он добавляется в систему с определенными привилегиями. Пользователям с привилегиями «Оператор» доступны все функции модуля «Оператор», как и пользователям с привилегиями «Преподаватель» доступны все функции модуля преподаватель.

Однако, у преподавателей существуют ограничения на доступ к некоторым данным. Например, преподаватель не может просмотреть журнал группы, у которой он не ведет занятия. Так же преподаватель не может изменить информацию о занятии, если ее вносил другой преподаватель.

На рисунке 8, изображена схема интерфейса информационной системы.

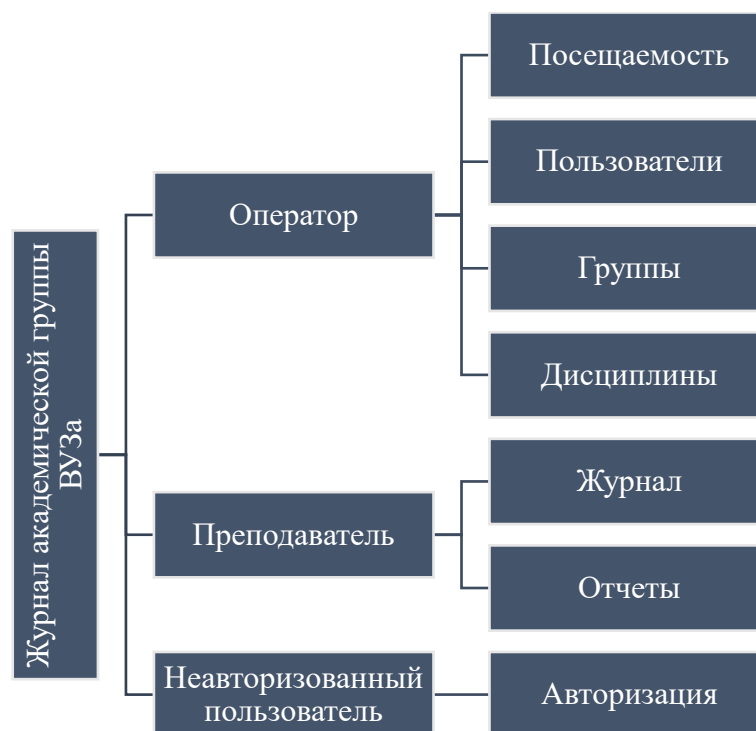


Рисунок 8 - Схема интерфейса информационной системы

На рисунке 8 видно, что выделяются три группы пользователей, каждой из которых доступны соответствующие страницы. Опишем назначение каждой из страниц:



- «Посещаемость» – на этой странице пользователь (оператор) может просмотреть информацию о посещаемости занятий в любой группе и за любой день.
- «Пользователи» – данная страница позволяет администратору управлять списком учетных записей информационной системы и осуществлять выгрузку отчетов по необходимым критериям.
- «Группы» – реализует управление списком академических групп, со всеми зависимыми параметрами (учебная нагрузка, подгруппы, распределение студентов по подгруппам), сортировка и фильтры отбора, выгрузка информации.
- «Дисциплины» – обеспечивает управление спискам дисциплин.
- «Журнал» – заполнение информации о занятиях и посещаемости студентов, сортировка записей, выгрузка информации о посещаемости.
- «Отчеты» – выгрузка месячной ведомости.
- «Авторизация» – форма авторизации в системе с проверкой капчи и функцией сброса пароля.

## **2.4 Реализация модулей информационной системы**

На этом этапе предстоит реализовать модули информационной системы. Для начала следует обеспечить возможность «наполнения» системы данными необходимыми для ее функционирования.

Часть данных будет содержать в «справочниках», например, форма обучения, пользовательские роли, информация о времени занятий, ученые степени и звания, факультеты, специальности, типы занятий и т. д., другая же часть должна поступать в систему от оператора или преподавателя.

На основе требований к информационной системе (Глава 1 раздел 2) преподаватель не может вносить сведения о студентах, дисциплинах или

группах, так как эти обязанности лежат на операторе. Иными словами, без рабочего модуля «Оператор», делать модуль «Преподаватель» невозможно.

Разработка модуля «Оператор» будет производиться поэтапно. Для начала создадим submodule «Пользователи». Этот submodule отвечает за управление учетными записями информационной системы.

В разделе 1.3 приводилось описание слоев информационной системы, где был выделен слой «Administrator», который позволяет хранить информацию об ученых записях, ролевой политике и хеш-коды сбросов пароля. Также был выделен слой «Users», позволяющий описать пользователя системы с учетом наиболее важных для реализации свойств.

Submodule «Пользователь» будет активно взаимодействовать именно с, описанными выше, слоями.

Определим основные функции submodule «Пользователь»:

1. Добавление студентов и преподавателей.
2. Редактирование студентов и преподавателей.
3. Удаление студентов и преподавателей.
4. Отображение информации о пользователях.
5. Фильтрации и поиск пользователей.
6. Выгрузка информации в EXCEL, PDF и csv или печать таблицы.

Реализация каждой выделенной функции требует соблюдения следующих правил:

1. Обязательный контроль входной информации: необходимо проверять все поступающие данные перед отправкой их в базы.
2. Минимизация ввода: требуется минимизировать ввод там, где это можно сделать, заменив его на поля выбора.

Теперь рассмотрим подробно процесс добавления нового пользователя в систему. Отметив, что контроль данных преимущественно возложен на клиентскую часть и заострять на нем внимание мы не будем.

Процесс добавления пользователя состоит из следующих этапов:

1. Запись информации в таблицу Users. На этом этапе в базу записывается служебная информация необходимая для идентификации пользователя и его ролевой политики: логин, пароль, email и роль.
2. В зависимости о роли создаваемого пользователя вносится информация о личных данных. Для студентов в таблицу StudentPprofile будет записано: фамилия, имя, отчество, дата рождения, фамилия и инициалы, пол, номер зачетной книжки, тип обучения (бюджет или коммерция), код группы; для преподавателей будет записано: фамилия, имя отчество, фамилия и инициалы, пол, дата рождения, должность, звание и кафедра.
3. Если все предыдущие стадии прошли успешно, то генерируется специальный код, которой заносится в таблицу AccauntActivation и связывается с идентификатором учетной записи.
4. Затем генерируется письмо со ссылкой, которое отправляется на указанный при добавлении адрес электронной почты.

Успешно созданный аккаунт не является активным и через него невозможно производить какие-то либо операции, до тех пор, пока пользователь, на чей адрес электронной почты было отправлено письмо, не перейдет по ссылке и не создаст свой пароль для авторизации. В разделе 1.4 уже приводилось описание процесса активации аккаунта.

Процесс редактирования данных выглядит аналогично, за исключением того, что перед редактированием необходимо получить имеющуюся информацию.

Для отображения информации о пользователях будем использовать таблицы (Рисунок 9). Для создания такой таблицы не понадобится использовать библиотеку DataTable, описанную в разделе 1.4.

The screenshot shows a web interface for a student table. At the top, there is a search bar labeled 'Поиск:' and a button 'Add Condition'. Below these are buttons for 'Печать' (Print), 'Выбрать столбцы' (Select columns), and a 'Показать' (Show) dropdown set to '5' with the text 'записей' (records). The table itself has columns: 'Фамилия' (Surname), 'Имя' (Name), 'Отчество' (Patronymic), 'Пол' (Gender), and 'Дата рождения' (Date of birth). It contains five rows of student data. Each row has a red trash icon and a blue edit icon. At the bottom, there is a pagination bar showing 'Записи с 1 до 5 из 12 записей' (Records from 1 to 5 of 12 records), buttons for 'Предыдущая' (Previous), '1', '2', '3', and 'Следующая' (Next).

Фамилия	Имя	Отчество	Пол	Дата рождения
Куликов	Константин	Петрович	М	1998-10-13
Петров	Петр	Петрович	М	1999-11-12
Макаров	Максим	Олегович	Ж	1999-11-23
Пупкин	Василий	Васильевич	М	1999-05-21
Ильин	Игорь	Андреевич	М	2021-01-15

Рисунок 9 - Таблица студентов

Библиотека DataTables позволяет представить таблицы как объект JavaScript, что упрощает управление ими. Как видно из рисунка 9, для таблицы реализованы возможности фильтрации, поиска, сортировки и печати данных. В дальнейшем такие таблицы будут использоваться во всех местах информационной системы, где есть необходимость представить данные в табличном виде.

Пример проверки корректности данных приведен на рисунке 10. Как видно из рисунка для полей ввода данных, на клиентской части, установлены специальные атрибуты, запрещающие пропускать или вводить данные, не характерные для поля.

The screenshot shows a form with two input fields. The first field contains '8y8y' and is highlighted with a blue border. The second field contains 'y87y'. Below the fields is a red error message with an exclamation mark icon: 'Адрес электронной почты должен содержать символ «@». Похоже, вы его пропустили: 8y8y.'

Рисунок 10 - Валидация почты

Следующим для реализации идет субмодуль «Группы» - самый большой и критически важный модуль информационной системы. Он отвечает за управление академическими группами. Под управлением предполагается создание, удаление и редактирование общей информации об академической

группе, а также настройка связанных параметров, таких как: нагрузка, подгруппы и списки студентов в общем для группы и для подгрупп.

Модуль разбит на блоки:

1. Создание группы – блок, отвечающий за добавление новой группы в базу данных
2. Таблица группы – блок отвечает за просмотр информации о группах в формате таблиц, поиск по ключевым характеристикам, сортировку и выгрузки информации в PDF, XLSX, печать. Данный блок реализован за счет библиотеки DataTables.
3. Конфигурация академической группы – блок отвечает за управление свойствами (полями) академической группы, а также разветвляется на подблоки, которые в свою очередь отвечают на настройки связанных параметров блока.

Рассмотрим подробно форму создания академической группы (рисунок 11). Из рисунка видно, что форма принимает девять параметров, однако, в структуре таблицы «AcademicGroups» полей восемь. Все дело в том, что поле «Факультет» служит фильтрующим полем для поля «Направление/Профиль». Выбирая нужный факультет, пользователю становится доступен список направлений, которые ведутся на данном факультет.

Код группы:	<input type="text"/>
Факультет:	<input type="text" value="Выберите нужное значение"/>
Направление/Профиль:	<input type="text" value="Выберите нужное значение"/>
Срок обучения:	<input type="text" value="Год начала"/> - <input type="text" value="Год окончания"/>
Текущий курс:	<input type="text" value="1"/>
Куратор группы:	<input type="text" value="Выберите нужное значение"/>
Форма обучения:	<input type="text" value="Выберите нужное значение"/>

Рисунок 11 - Форма создания академической группы

Такая реализация возможна с использованием AJAX о котором было сказано в разделе 2.2. Рассмотрим подробнее логику работы формы. Когда пользователь выбирает факультет браузер генерирует событие change. Это событие говорит, что в поле (в данном случае это select) произошли изменения и вызывает некую функцию обработчик (листинг 1), которая была «навешана» на это событие.

#### Листинг 1 - AJAX запрос списка направлений

```

$("#ListFaculty").change(function () { $.ajax({
    method: 'GET',
    url: 'API/getSpecialtyNameList.php',
    data: {id: $("#ListFaculty").val()},
    cache: false,
    success: function (json) {
        json = JSON.parse(json);
        $("#ListSpecialty").html('');
        for (let hID in json) {
            $("#ListSpecialty").append("<option
value='" + json[hID]['ID'] + "'><div>" + json[hID]['Name'] +
"</div></option>"); } } } } });

```

Разберем подробнее параметры AJAX:

1. Method – определяют метод передачи параметров серверу. В данной информационной системе используются методы GET и POST, о них было сказано в разделе 2.2.
2. url – адрес php-скрипта, который должен обработать полученный запрос;
3. data – параметры запроса. В случае с листингом 1, на сервер передается идентификатор факультета;
4. cache – параметр, говорящий AJAX, стоит ли кешировать ответ. Этот параметр используется, если заведомо известно, что данные, которые должны прийти от сервера, не будут часто изменяться;
5. Success – в случае успеха AJAX будет выполнена функция с аргументами в виде данных ответа сервера. В данном случае, получив список специальностей, функция заполняет поле ввода select, которое отвечает за выбор специальностей.

Отметим, что данные, заполняемые в форме добавления академической группы (рисунок 11), также отправляются на сервер с помощью AJAX технологии.

Таблица со списком групп оформляется аналогично таблице пользователей. Однако данные для нее берутся с помощью AJAX запроса, что позволяет упростить реализацию серверной части. Отметим, что записи таблицы выступают в роли кнопок, при нажатии на которые пользователю открывается интерфейс конфигурации академической группы.

Блок конфигурации академической группы отвечает за настройку параметров группы и разделяется на несколько подблоков.

Первый подблок «Конфигурация» реализует функциональность редактирования общих сведений о группе. При открытии формы с помощью

AJAX-запроса из базы данных подгружается информация об академической группе. Форма редактирования выглядит аналогично форме создания академической группы.

Следующим подблоком является подблок «Студенты», отвечающий за редактирование информации о студентах выбранной группы. Этот подблок аналогичен по функционалу субмодулю студенты, с тем лишь отличием, что ограничивает выборку студентов в пределах одной академической группы.

Подблок «Дисциплины» (рисунок 12), реализует управление нагрузкой академической группы.

**Учебная нагрузка группы**

Здесь Вы можете назначить список дисциплин для академической группы.

[Добавить нагрузку](#)

Фильтр Print Excel Column visibility Поиск:

Дисциплина	Преподаватель	Тип занятий	Семестр	Количество часов
Программирование	Архипов Вадим Олегович	Лекция	7	36
Программирование	Прилепин Август Иванович	Практическое занятие	7	18
Программирование	Архипов Вадим Олегович	Практическое занятие	7	18
Программирование	Архипов Вадим Олегович	Экзамен	6	2

Записи с 1 до 4 из 4 записей

Предыдущая 1 Следующая

Рисунок 12 - Подблок "Дисциплины"

Стоит обратить внимание, что для каждой группы должна быть создана учебная нагрузка. Это нагрузка определяет список занятий со всеми зависимыми параметрами: преподаватель, тип занятия, семестр, количество часов, форма контроля. Именно через учебную нагрузку преподавателю назначается ведение дисциплины в конкретной группе. Если такая нагрузка не назначена, то преподаватель не сможет получить доступ к информации дисциплине в выбранной группе.



Как видно из рисунка 12, таблица дисциплин позволяет импортировать информацию, выполнять поиск, сортировку и фильтрацию данных.

Рассмотрим подробнее функционал, который предоставляет Data Tables и который реализован в большинстве таблицы информационной системы на примере таблицы дисциплин.

Например, необходимо выгрузить в XLSX -файл (т. е. Excel) информацию о дисциплинах за седьмой семестр в группе 120871. Информация должна быть представлена в следующем формате:

1. Дисциплина.
2. Преподаватель.
3. Тип занятий.
4. Количество часов.

Для отбора информации используется составной фильтр (Рисунок 13). Фильтры, реализуемые библиотекой DataTables, позволяют создавать сложные условия отбора данных.



Рисунок 13 - Фильтр записей

После фильтрации необходимо выбрать поля, которые будут использованы для выгрузки. Для этого необходимо нажать на кнопку «Column Visibility» и отметить интересующие нас столбцы. Таблица дисциплин примет вид, как показано на рисунке 14. Как видно из рисунка, пропали поля, содержащие информацию о семестре, форме контроля и дополнительной нагрузке.

Дисциплина ▲	Преподаватель ◆	Тип занятий ◆	Количество часов ◆
Программирование	Архипов Вадим Олегович	Лекция	36
Программирование	Прилепин Август Иванович	Практическое занятие	18
Программирование	Архипов Вадим Олегович	Практическое занятие	18

Рисунок 14 - Таблица дисциплин после фильтрации

После нажатия кнопки «Excel» браузер предложит пользователю сохранить информацию на жесткий диск компьютера. В качестве информации подразумевается XLSX-файл, содержащий отфильтрованные данные (Рисунок 15).

Учебная нагрузка			
Дисциплина	Преподаватель	Тип занятий	Количество часов
Программирование	Архипов Вадим Олегович	Лекция	36
Программирование	Прилепин Август Иванович	Практическое занятие	18
Программирование	Архипов Вадим Олегович	Практическое занятие	18

Рисунок 15 - XLSX-файл с отфильтрованными данными

Такая выгрузка реализована для большинства таблиц в системе и заострять на ней внимание в дальнейшем мы не будем. Все скриншоты и исходные коды будут приведены в приложении.

После добавления некой дисциплины в список нагрузки академической группы, пользователю становятся доступны функции управления подгруппами. Для создания подгруппы необходимо выбрать учебную нагрузку и указать номер подгруппы (Рисунок 16).

Рисунок 16 - Создание подгруппы

После создания подгруппы она отображается в таблице подгрупп (Рисунок 17). Как видно из рисунка, подгруппы создаются для каждого вида учебной нагрузки.

Дисциплина ▲	Преподаватель ▼	Тип занятий ▼	Номер подгруппы ▼
+ Программирование	Архипов Вадим Олегович	Лекция	2
+ Программирование	Прилепин Август Иванович	Практическое занятие	1
+ Программирование	Прилепин Август Иванович	Практическое занятие	2

Рисунок 17 - Таблица подгрупп

Таблица подгрупп, как и все другие таблицы информационной системы, реализована с помощью библиотеки DataTables. На строки таблицы наложен обработчик события click (одинарное нажатие левой кнопки мыши), который вызывает функцию для отображения формы управления списком студентов в подгруппе (Рисунок 18). Такая форма реализована при помощи DataTables, с активированным свойством multiselect. При нажатии кнопки «Добавить» формируется массив с идентификаторами студентов, затем, через AJAX-запрос, на сервер передается сформированный массив и идентификатор подгруппы. Аналогичным образом построена логика работы кнопки «Исключить»

Добавить
Поиск:

**Все студенты** ▲

- ☐ Петров Петр Петрович
- ☐ Рахманов Илья Олегович
- ☐ Филонов Максим Андреевич
- ☐ Шукин Аркадий Николаевич

Исключить
Поиск:

**Студенты данной подгруппы** ▲

- ☐ Антонов Николай Иванович
- ☐ Куликов Константин Петрович
- ☐ Пупкин Василий Васильевич

Назад
Удалить подгруппу

Рисунок 18 - Форма управления списком студентов подгруппы

Также в блоке «Конфигурация» доступна опция удаления академической группы. Здесь стоит обратить внимание на то, что при

удалении академической группы уничтожается вся информация, связанная с ней. А именно:

1. Сведения об учебной нагрузке.
2. Информация о подгруппах.
3. Все студенты группы.
4. Информация о проведенных занятиях.
5. Данные о посещаемости занятий.

Такое поведение функции удаления академической подгруппы обусловлено разработанной в разделе 1.3 структуре базы данных, а именно за счет установленных связей между сущностями.

Для исключения ситуации случайного удаления группы был создан механизм, который требует, чтобы оператор ввел код группы, которую он желает удалить в окне конфигурации группы (Рисунок 19). Пока код группы не введен или указан неверно, кнопка «Удалить» является не активной.

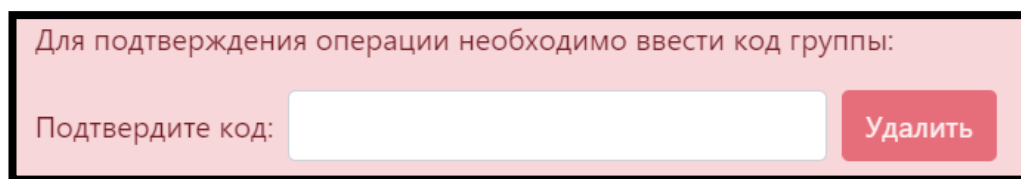


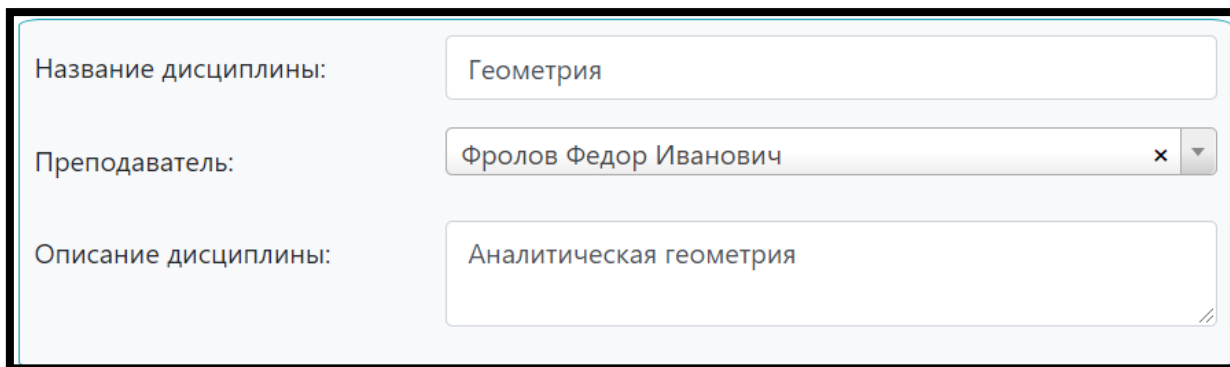
Рисунок 19 - Подтверждение кода группы

После нажатия кнопки «Удалить» на сервер, специальному скрипту отправляется AJAX-запрос с кодом группы. Если операция удаления на сервере прошла успешно, то пользователь увидит сообщение об успехе, в противном случае – ему будет показано сообщение об ошибке с информацией.

Заключительным этапом разработки модуля «Оператор» является создание субмодуля «Дисциплины». Этот субмодуль находится на слое «Reference» и заполняется оператором информационной системы.

На рисунке 20, показана форма добавления/редактирования дисциплины. Как и многие другие формы данной информационной системы, форма добавления и редактирования работает с данными через AJAX.

Единственным отличием является то, что для добавления и редактирования используется один объект.



Название дисциплины: Геометрия

Преподаватель: Фролов Федор Иванович x

Описание дисциплины: Аналитическая геометрия

Рисунок 20 - Форма добавления/редактирования дисциплин

На следующем этапе создания информационной системы предстоит разработать модуль «Преподаватель». На основании требований, описанных в разделе 1.3 этот модуль разбивается на субмодули: «Занятия» и «Отчеты».

Субмодуль «Занятия» реализует следующую функциональность:

1. Запись информации о занятиях и посещаемости.
2. Редактирование информации о занятиях и посещаемости.
3. Удаление информации о занятиях и посещаемости.
4. Отображение информации о занятиях и посещаемости.
5. Выгрузка отчета о посещаемости.

Субмодуль «Отчеты» реализует выгрузку месячной ведомости учета работы профессорско-преподавательского состава.

Форма добавления занятий состоит из двух основных частей:

1. Информация о занятии.
2. Посещаемость.

Информация о занятии (Рисунок 21) описывает занятие следующими свойствами: дата проведения занятия, название дисциплины (она автоматически подставляется, когда пользователь находится в журнале какой-либо группы), тип занятия (практическое, лекция, лабораторные и т. д.), тема занятия, номер пары на который проходило это занятие.

Дата	<input type="text" value="03.06.2021"/>
Дисциплина:	<input type="text" value="Программирование"/>
Тип занятия:	<input type="text" value="Выберите значение"/>
Тема занятия	<input type="text"/>
Номер пары:	<input type="text" value="1 пара (08:40 - 10:15)"/>

Рисунок 21 - Форма записи информации о занятии

Стоит отметить, что для преподавателя в поле «Тип занятия» доступны только те типы, которые были указаны оператором информационной системы в учебной нагрузке группы.

После заполнения информации о занятии, преподаватель может внести данные о посещаемости (Рисунок 22).

Посещаемость			
Подгруппа:			
<input type="text" value="Все студенты"/>			
			Поиск: <input type="text"/>
Студент	Пропущено	Баллы	Комментарий
Антонов Николай Иванович	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text"/>
Куликов Константин Петрович	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text"/>

Рисунок 22 - Заполнение посещаемости

Преподавателю предлагается брать подгруппу или заполнить данные обо всех студентах. Стоит отметить, что список подгрупп доступен только если преподаватель выбрал тип занятий. Это связано с тем, что по умолчанию, если для учебной нагрузки не указаны подгруппы, то преподаватель видит всех студентов.

Как видно из рисунка 22, преподаватель для каждого студента заполняет три поля:

1. «Пропущено» – здесь преподаватель отмечает количество пропущенных студентом часов. 0 – студент был на паре, 1 – студент пропустил первую половину пары, 2 – студент пропустил полностью пару.
2. «Баллы» – преподаватель может поставить здесь баллы, которые заработал студент на занятии.
3. «Комментарий» – это не обязательное поле, в которое преподаватель может вписать свой комментарий о студенте.

Логика добавления занятия состоит в следующем:

1. На клиентской части формируются четыре массива с описанием полей посещаемости (идентификаторы студентов, сколько часов, баллы и комментарий).
2. На сервер с помощью AJAX – запроса отправляется информация о занятии и посещаемости.
3. Серверный скрипт создает занятие.
4. Если занятие успешно создано, то проходит циклом по массиву и записывает информацию о посещаемости.
5. Если занятие создать не удалось, то сообщает об этом пользователю.

Алгоритм редактирования занятия выглядит аналогично, за тем лишь исключением, что для заполнения данных о посещаемости подгружается список студентов, которые были ранее указаны при создании занятия. Иными словами, если при создании занятия была выбрана первая подгруппа, то и при редактировании отобразиться список студентов из первой подгруппы и расширить его уже будет невозможно. Такое поведение обусловлено тем, что в базе данных, а именно в таблице `attendacelesson`, хранится информация в виде: идентификатор занятия, идентификатор студента и сведения из блока

посещаемости. Из этого следует что, если в таблице за конкретное занятие нет информации о студенте — это значит, что он либо на момент создания занятия не числился в этой группе, либо был в другой подгруппе.

Для просмотра журнала преподавателю необходимо выбрать группу и дисциплину (Рисунок 23), которую он ведет в этой группе. Стоит отметить, что в списках выбора дисциплин, групп и типов занятий, преподаватель может видеть только те параметры, которые были заданы через учебную нагрузку.

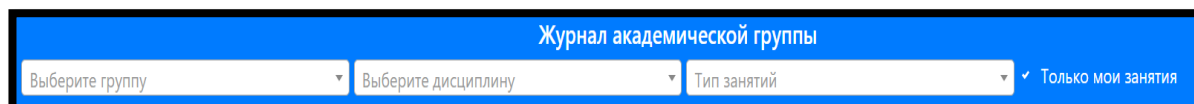


Рисунок 23 - Форма выбора параметров для загрузки журнала

Как видно из рисунка 23, помимо выбора группы и дисциплины, преподаватель также может отфильтровать журнал группы по типу занятий, а также просмотреть все занятия, включая те, которые он не ведет, однако редактирование их ему недоступно (Рисунок 24).

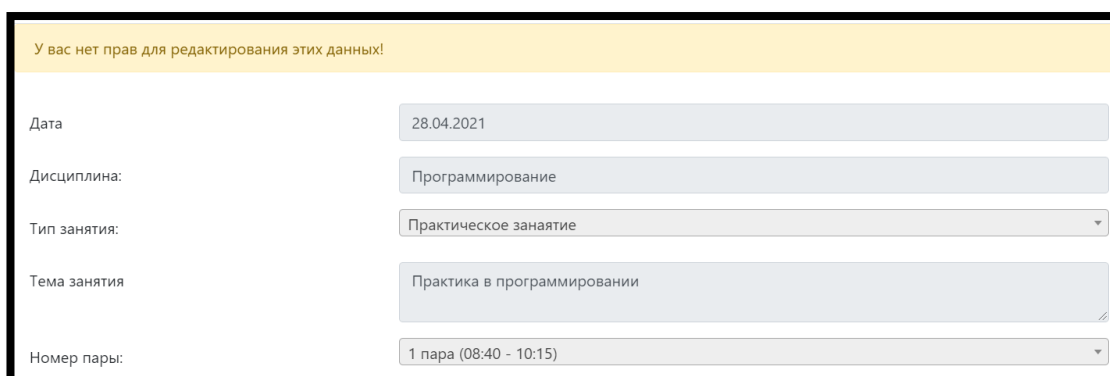


Рисунок 24 - Просмотр информации о занятии другого преподавателя

На рисунке 25 показана таблица посещаемости студентов. Данная таблица имеет сложную структуру для построения, поэтому, в отличие от большинства таблиц информационной системы, она генерируется на сервере и там же формируется объект DataTables.



Студенты	01.04.21	27.04.21	20.05.21	21.05.21	21.05.21	03.06.21	Пропущено часов	Баллы
	ЛК	ЛК	ЛК	ЛК	ЛК	ЛК		
Антонов Николай Иванович	2 (Был)	0 (Н/Б)	12 (Был)	5 (Был)	-2 (Н/Б)	0 (Н/Б)	6	17
Куликов Константин Петрович	2 (Был)	2 (Был)	7 (Был)	-2 (Н/Б)	-2 (Н/Б)	0 (Н/Б)	6	7
Петров Петр Петрович	2 (Был)	-2 (Н/Б)	13 (Был)	8 (Был)	-2 (Н/Б)	0 (Н/Б)	6	19
Пупкин Василий Васильевич	2 (Был)	2 (Был)	15 (Был)	-2 (Н/Б)	7 (Был)	0 (Н/Б)	4	24
Рахманов Илья Олегович	2 (Был)	2 (Был)	12 (Н/Б)	6 (Был)	6 (Был)	0 (Н/Б)	4	28
Филонов Максим Андреевич			17 (Был)	-2 (Н/Б)	-2 (Н/Б)	0 (Н/Б)	6	13
Щукин Аркадий Николаевич	2 (Был)	2 (Был)	11 (Н/Б)	4 (Был)	4 (Был)	0 (Н/Б)	4	23
<div>Copy Excel CSV PDF Print</div>								

Рисунок 25 - Таблица посещаемости занятий

Таблица, изображенная на рисунке 25 имеет следующую структуру:

1. Фиксированный столбец «Студенты» содержит список студентов и поддерживает сортировку.
2. За столбцом студентов следует динамически изменяемая цветная область данных о посещаемости, не являющееся фиксированной относительно всей таблицы.
3. Затем следуют два столбца с информацией об общем количестве пропусков и накопленных баллов. Эти столбцы фиксированы и поддерживают сортировку.

Из представленной выше структуры следует что область с информацией о посещаемости имеет горизонтальную прокрутку между столбцами «Студенты» и «Пропущено часов». Такой подход был обусловлен тем, что при большом числе занятий таблица посещаемости была бы трудно читаема.

Цвета таблицы обозначают количество пропущенных часов: красный – 2 часа, желтый – 1 час, зеленый – 0 часов, а серым отмечены студенты, о которых нет информации в таблице посещаемости Attendacelesson.

Также, как видно из рисунка 25, таблица посещаемости занятий поддерживает выгрузку данных в наиболее востребованные форматы, а также печать.

Логика формирования таблицы посещаемости без использования фильтров «Тип занятия» и «Только мои занятия» выглядит следующим образом:

1. На сервер передается код группы, идентификатор дисциплины и идентификатор преподавателя.
2. Скрипт на сервере формирует шапку таблицы, в которую входят даты проведения занятий.
3. Если «шапка дат» сформирована, то система запрашивает список студентов данной подгруппы.
4. Затем для каждого студента, по его идентификатору, запрашивается информация для каждого занятия. В процессе запросов формируется таблица.
5. Результирующая таблица возвращается в ответ на AJAX-запрос.

При использовании фильтров логика выглядит также, только параметров серверу передается больше.

Заключительным этапом разработки модуля «Преподаватель» является реализация субмодуля «Отчеты». Данный субмодуль отвечает за выгрузку месячной ведомости учета работы профессорско-преподавательского состава в файл XLSX. Для реализации подобной выгрузки недостаточно возможностей DataTables, поэтому было использована библиотека PhpSpreadshee, описанная в разделе 1.4.

На рисунке 26 представлен интерфейс модуля выгрузки отчетов. Преподаватель может получить отчет только за конкретный месяц.

**Месячная ведомость**

Май

[Сохранить в Excel](#)

Дата	Часы занятий	Факультет	Курс	Группа	Содержание занятий	Вид занятий	Количество часов
2021-05-01	08:40-10:15	МФИИ	1	127987-А	Лекция по теме 1	ЛК	2
2021-05-01	10:25-12:00	МФИИ	1	127987-А	Лабораторная работа №1	ПЗ	2
2021-05-08	08:40-10:15	МФИИ	1	127987-А	Лекция по теме 2	ЛК	2
2021-05-08	10:25-12:00	МФИИ	1	127987-А	Лабораторная работа №2	ПЗ	2

Рисунок 26 - Интерфейс модуля выгрузки отчетов

После нажатия кнопки «Сохранить в Excel» на сервер с помощью AJAX отправляется идентификатор преподавателя и номер месяца. Серверная часть, получив данные, начинает сбор следующей информации:

1. Общие сведения о преподавателе (факультет, кафедра, фамилия, имя и отчество, должность, ученое звание).
2. Информация о проведенных занятиях (дата, часы занятий, факультет, курс, группа, темы занятий, тип занятий).

Если по какой-либо причине сервер не смог получить какие-либо данные, то пользователь получит об этом уведомление и выгрузка будет прервана.

В случае успеха, на компьютер пользователя будет загружен XLSX файл с информацией о проведенных занятиях за выбранный месяц (Рисунок 27).

Тульский государственный педагогический университет им. Л.Н. Толстого										УТВЕРЖДЕНА Министерством образования РФ 24.01.98 г.								Форма №2	
МЕСЯЧНАЯ ВЕДОМОСТЬ																			
учета работы профессорско-преподавательского состава																			
за МАЙ 2021 г.																			
Факультет		Математики физики и информатики						Кафедра		Информатики и информационных технологий									
Ученое звание и степень						д. ф.-м. н, профессор, Архипов Вадим Олегович													
						Количество выполненных часов по видам занятий												Подпись преподавателя	
Дата	Часы занятий	Факультет	Курс	Группа	Содержание занятий	Лекции	Практ. занятия	Консультации	Зачеты	Экзамены	Руков. пед. практик.	Руков. курс. и диал. раб.	Вычислит. практ.	КР	КСРС	ИГА			
20.05.21	08:40-10:15	МФИИ	4	120871	Лабораторная работа 2. Указатели		2												
20.05.21	10:25-12:00	МФИИ	4	120871	Динамические структуры данных	2													
20.05.21	12:40-14:15	МФИИ	1	127987-А	Функции пользователя	2													
20.05.21	14:25-16:00	МФИИ	4	120871	Библиотека математических функций		2												

Рисунок 27 - Месячная ведомость

Таким образом, в данном разделе были рассмотрены основные этапы разработки информационной системы «Журнал академической группы». Все исходные коды можно будут посмотреть в приложении.

## **2.5 Тестирование информационной системы**

Тестирование является важным этапом разработки любой информационной системы, целью которого является выявление «дефектов» в работе системы, прежде чем она будет допущена к конечному пользователю.

Для данной информационной системы проводилось Альфа-тестирование, целью которого являлся поиск и устранение ошибок в процессе разработки. Тестированию подвергались как критичные модули системы, так и общие модули взаимодействия с пользователем. В ходе тестирования выполнялась проверка следующих частей информационной системы:

1. Корректность обработки данных: проверка правильности ввода дат и корректной реакции системы на неправильные параметры.
2. Непротиворечивость структуры базы данных: не нарушает ли схема базы данных общую логику работы информационной системы «Журнал академической группы».
3. Правильность работы системы при добавлении и удалении тестовых данных.
4. Проверка корректности SQL-запросов.
5. Проверка правильности генерации отчетов.
6. Общее тестирование интерфейса информационной системы.

В ходе проведения тестирования были обнаружены и устранены следующие ошибки:

1. На некоторых формах ввода не выполнялась проверка корректности данных. Это было связано с тем, что кнопка, отвечающая за отправку данных, не была связана с формой ввода.

2. Ошибка выборки для студентов для списка подгрупп: некоторые студенты не отображались в списке студентов, не вошедших в подгруппу. Ошибка связана с конфликтом записей в базе данных и была устранена путем изменения SQL-запроса.

3. Незначительные ошибки и уязвимости в информационной системе.

В ходе альфа-тестирования иных ошибок не обнаружено. Однако, следует принимать во внимание, что альфа-тестирование не гарантирует корректность работы информационной системы и потому требуется как минимум Бета-тестирование.

Идея Бета-тестирования предполагает, что информационная система будет тестироваться большим количеством людей (как правило это конечные пользователи или добровольцы, которые хотят помочь развитию проекта), знающими предметную область.

В процессе разработки информационной системы использовалась система контроля версий Git и сервис хостинга ИТ-проектов GitHub. Все исходные коды с последними изменениями доступны на портале GitHub в открытом репозитории под лицензией Apache License 2.0. Ссылка на репозиторий будет приведена в описании.

## **2.6 Развертывание информационной системы**

Информационная система «Журнал академической группы ВУЗа» имеет следующие требования для установки:

1. Http – сервер Nginx версии 1.14
2. PHP версии 7.1
3. MariaDB 10.3

Для воссоздания, описанных выше требований, можно использовать пакет Open Server версии 5.2.2. Скачать пакет и инструкцию по установке можно с официального сайта проекта Open Server: <https://ospanel.io/>

После установки пакета необходимо выполнить начальную конфигурацию сервера:

1. Запустите Open Server;
2. На панели задач найдите флажок (Рисунок 28) (сразу после запуска он будет красного цвета, это означает, что сервер не запущен еще) и нажмите на него.

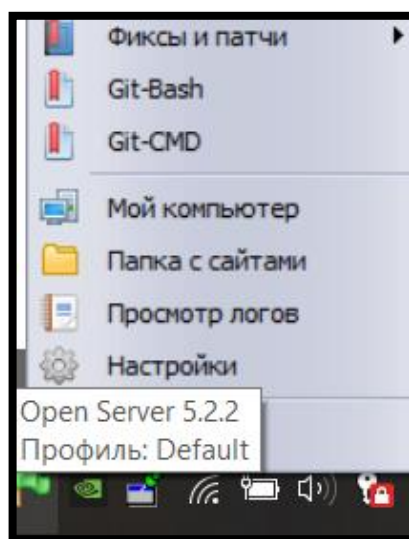


Рисунок 28 - меню Open Server

3. В раскрывавшемся меню выберите настройки
4. После того, как откроется окно настроек, перейдите во вкладку «Модули» (рисунок 29) и выполните настройку модулей согласно описанным выше правилам.
5. После завершения этапа настройки в меню Open Server нажмите кнопку «Запустить».
6. Если значок Open Server в панели задач загореться зеленым, значит сервер запущен и можно переходить к этапу экспорта информационной системы.

Стоит отметить, что установленные на данном этапе параметры не следует менять в дальнейшем, так как для каждого вида параметров

необходима настройка дальних параметров (файлы информационной системы и схема баз данных).

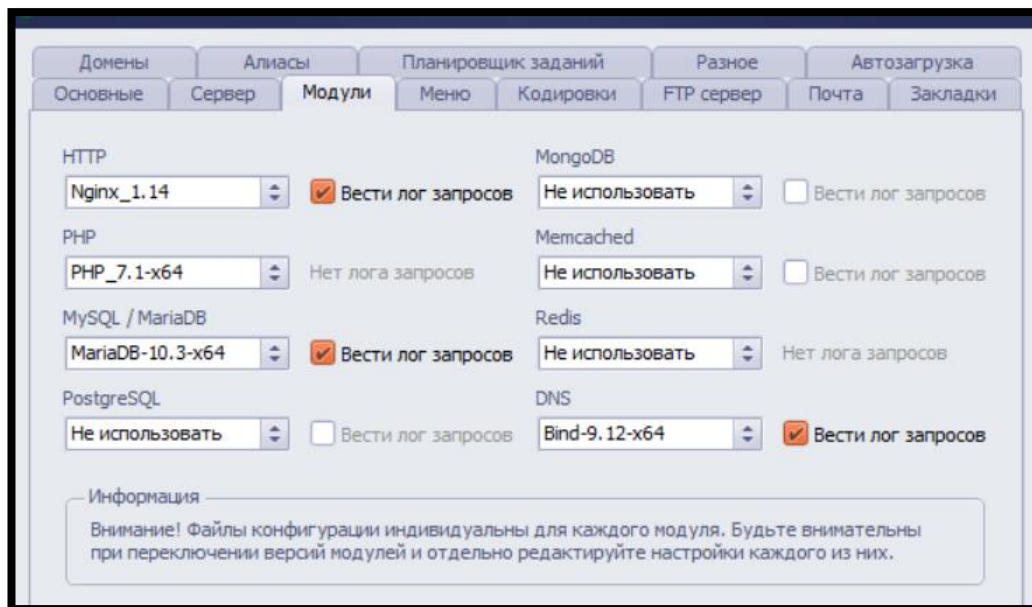


Рисунок 29 - Интерфейс настройки Open Server

Как было сказано ранее все исходные коды информационной системы «Журнал академической группы ВУЗа» находятся в открытом доступе в репозитории GitHub.

Для получения исходных кодов информационной системы есть два способа:

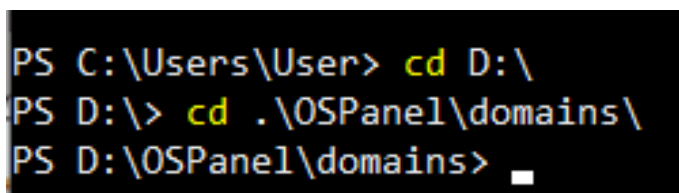
1. Клонирование репозитория в каталог domains. Domains – является рабочим (корневым) каталогом сервера и все сайты располагаются именно в нем.
2. Скачивание zip-архива с исходными кодами и распаковка архива в рабочий каталог Domains.

Рассмотрим подробнее первый шаг. Для ее реализации необходимо, чтобы на компьютере пользователя была установлена утилита Git.

Для «клонирования» репозитория необходимо:

1. Открыть терминал (командная строка, в данном примере будет использован терминал PowerShell)

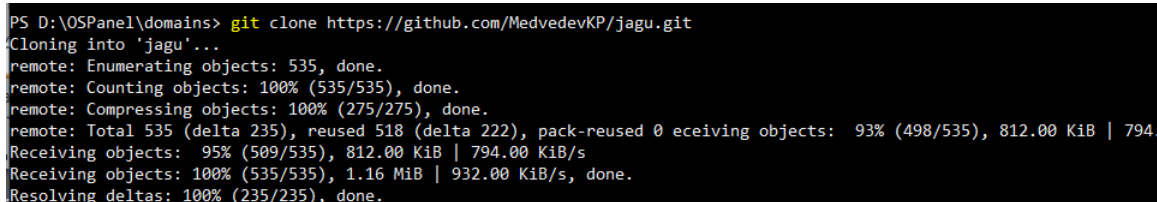
2. Перейти рабочий каталог Open Server domains (Рисунок 30)



```
PS C:\Users\User> cd D:\
PS D:\> cd .\OSPanel\domains\
PS D:\OSPanel\domains> _
```

Рисунок 30 - переход в рабочий каталог Open Server

3. После перехода в рабочий каталог необходимо выполнить команду для клонирования репозитория: `git clone https://github.com/MedvedevKP/jagu.git`
4. После нажатия Enter начнется скачивание репозитория в папку domains



```
PS D:\OSPanel\domains> git clone https://github.com/MedvedevKP/jagu.git
Cloning into 'jagu'...
remote: Enumerating objects: 535, done.
remote: Counting objects: 100% (535/535), done.
remote: Compressing objects: 100% (275/275), done.
remote: Total 535 (delta 235), reused 518 (delta 222), pack-reused 0
Receiving objects: 95% (509/535), 812.00 KiB | 794.00 KiB/s
Receiving objects: 100% (535/535), 1.16 MiB | 932.00 KiB/s, done.
Resolving deltas: 100% (235/235), done.
```

Рисунок 31 - результат клонирования репозитория

На этом клонирования репозитория завершен. В папке domain появится папка jagu со всеми необходимыми модулями. Завершающим этапом инсталляции системы является импорт базы данных.

Для импорта базы данных откройте меню Open Server (через значок на панели задач) и выберите «Дополнительно/phpMyAdmin» (Рисунок 32). После нажатия на пункт «phpMyAdmin» откроется окно браузера с панелью администратора базы данных. Изначально будет запрошен логин и пароль. Эти параметры определяются через конфигурацию Open Server и ПО умолчанию являются «Admin/Admin», однако все может зависеть от версии сборки Open Server.

После прохождения успешной авторизации откроется окно панели администратора баз данных (Рисунок 32).



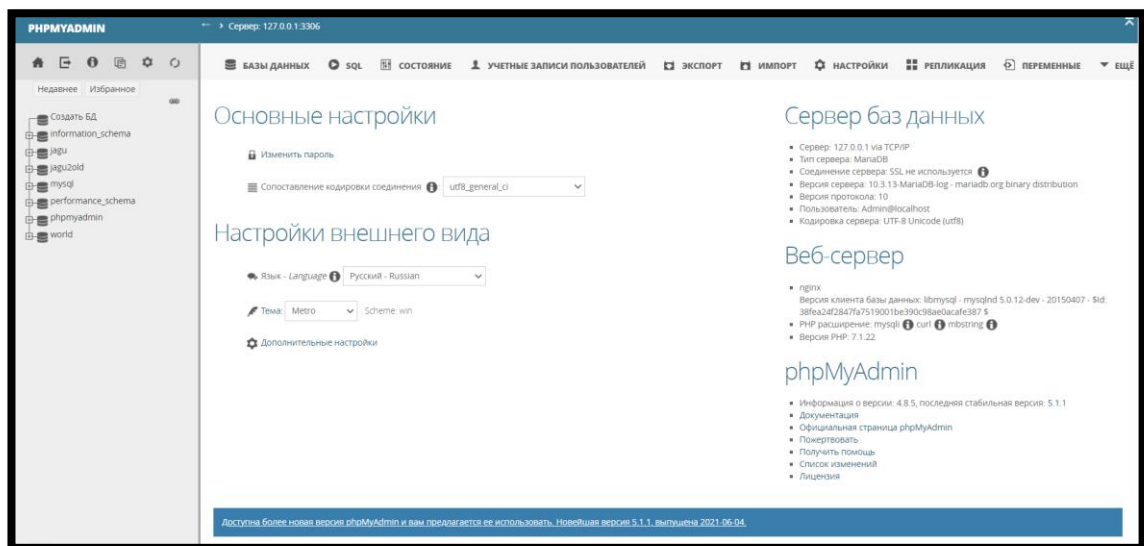


Рисунок 32 - Интерфейс Open Server

Затем необходимо перейти в раздел «Импорт» и выбрать пункт «Обзор вашего компьютера» и нажать кнопку «Выберите файл».

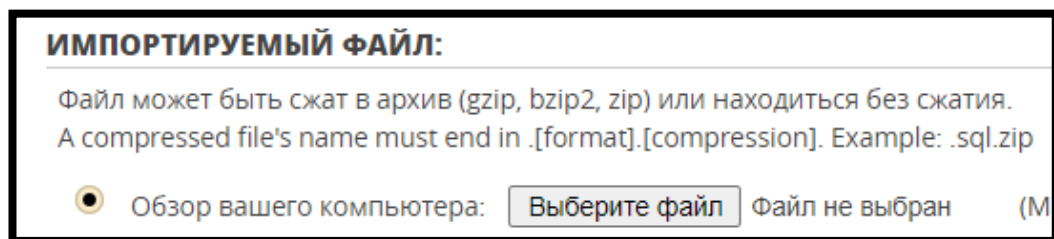


Рисунок 33 - Раздел импорта базы данных

В открывшемся окне выбора необходимо перейти к каталогу `jagu`, который был клонирован на предыдущих шагах. В каталоге `jagu` перейдите в папку `datatables` и выберите файл `jagu.sql`. После выбора файла в самом низу страницы будет кнопка «Вперёд», на нее нужно нажать.

В результате будет произведен импорт базы данных и будет выдано уведомление о результате импорта (рисунок 34)

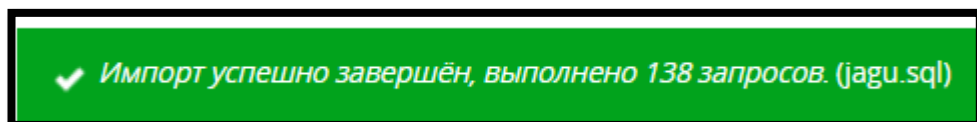


Рисунок 34 - результат импорта базы данных

Для запуска информационной системы необходимо открыть меню сервера и убедившись, что сервер запущен (отображается зеленый значок на панели задач), нужно выбрать и нажать на название информационной системы, в данном случае это jagu (Рисунок 35).

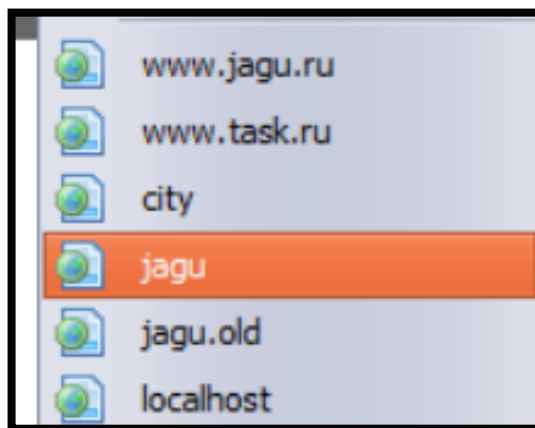


Рисунок 35 - список сайтов на сервере

В случае успеха – откроется окно браузера, где будет загружена страница авторизации информационной системы. В системе уже задан пользователь «Оператор» с логином admin и паролём admin. В дальнейшем эти параметры авторизации необходимо будет поменять в целях обеспечения безопасности.

## 2.7 Выводы по главе 2

1. Информационная система «Журнал академической группы ВУЗа» разделена на основные модули: «Оператор» и «Преподаватель».
2. В качестве архитектуры была выбрана «клиент-серверная» архитектура построения приложения.
3. Создана структура информационной системы с описанием функционала в каждой единице структуры.
4. Выполнена пошаговая реализация информационной системы «Журнал академической группы ВУЗа» с описание основных компонентов и модулей.

5. Проведено альфа-тестирование. Рассмотрены наиболее критические ошибки функционирования информационной системы «Журнал академической группы ВУЗа».
6. Разработана инструкция по развёртыванию информационной системы «Журнал академической группы ВУЗа» на локальной машине пользователя. Разобраны основные этапы конфигурации сервера и базы данных для нормальной работы информационной системы.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения выпускной квалификационной работы была разработана информационная система «Журнал академической группы ВУЗа».

Были решены следующие задачи:

1. Проведено исследование предметной области «Журнал академической группы ВУЗа».
2. Выполнено проектирование информационной системы, выделены основные сущности и связи между ними.
3. Выделены основные компоненты разработки: модули и субмодули информационной системы.
4. Проведена поэтапная разработка информационной системы «Журнал академической группы ВУЗа».

Помимо решения основных задач данной выпускной квалификационной работы, были также решены следующие подзадачи:

1. Рассмотрены этапы разработки информационной системы, что позволяет иметь общее представление о процессе создания информационных систем.
2. Определены требования к информационной системе «Журнал академической группы ВУЗа», что позволяет наложить ограничение и выделить границы детализации предметной области.
3. С учетом определенных требований было проведено моделирование информационной системы и выделены основные сущности.
4. Рассмотрены средства и технологии разработки, которые были задействованы в ходе реализации информационной системы «Журнал академической группы ВУЗа».
5. Описана архитектура построения информационной системы.

6. Описана структура информационной системы. Проведено альфа-тестирование, описаны и исправлены основные ошибки функционирования информационной системы «Журнал академической группы ВУЗа».

Таким образом, были решены все задачи, которые описаны во введении, цель достигнута.

В дальнейшем информационная система может быть функционально расширена, что поддерживается структурой базы данных. Как было быше все исходные коды данной системы доступны в публичном репозитории.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Аврунев, О.Е. Модели баз данных : учебное пособие : [16+] / О.Е. Аврунев, В.М. Стасышин – Новосибирск : Новосибирский государственный технический университет, 2018. – 124 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=575324> (дата обращения: 17.12.2020). – Библиогр. в кн. – ISBN 978-5-7782-3749-0. – Текст : электронный.
2. Гущин, А.Н. Базы данных : учебник / А.Н. Гущин. – Москва : Директ-Медиа, 2014. – 266 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=222149> (дата обращения: 17.12.2020). – ISBN 978-5-4458-5147-9. – DOI 10.23681/222149. – Текст : электронный.
3. Дьяков, И.А. Базы данных. Язык SQL : учебное пособие / И.А. Дьяков – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2012. – 82 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=277628> (дата обращения: 17.12.2020). – Библиогр. в кн. – Текст : электронный.
4. Методы анализа предметных областей : учебно-методический комплекс / сост. И.Л. Скипор – Кемерово : Кемеровский государственный университет культуры и искусств (КемГУКИ), 2014. – 76 с. : табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=275378> (дата обращения: 17.12.2020). – Текст : электронный.
5. Строганов, А.С. Ваш первый сайт с использованием PHP-скриптов : учебное пособие / А.С. Строганов. – 3-е изд., испр. и доп. – Москва : Диалог-МИФИ, 2015. – 288 с. : ил. – Режим доступа: по подписке. –

- URL: <https://biblioclub.ru/index.php?page=book&id=447998> (дата обращения: 17.12.2020). – ISBN 978-5-86404-226-7. – Текст: электронный.
6. Туманов, В.Е. Основы проектирования реляционных баз данных : учебное пособие / В.Е. Туманов. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ): Бином. Лаборатория знаний, 2007. – 421 с. : ил., табл., схем. – (Основы информационных технологий). – Режим доступа: по подписке. –
- URL: <https://biblioclub.ru/index.php?page=book&id=233290> (дата обращения: 17.12.2020). – ISBN 978-5-9556-0111-3. – Текст : электронный.
7. Шабашов, В.Я. Организация доступа к данным из РНР приложений для различных СУБД: учебное пособие по дисциплине «Web-программирование» : [16+] / В.Я. Шабашов. – Москва ; Берлин : Директ-Медиа, 2019. – 121 с. : ил., табл. – Режим доступа: по подписке. –
- URL: <https://biblioclub.ru/index.php?page=book&id=499185> (дата обращения: 17.12.2020). – Библиогр.: с. 90. – ISBN 978-5-4475-9888-4. – DOI 10.23681/499185. – Текст: электронный.
8. ГОСТ 2.105-95 Общие требования к текстовым документам
9. ГОСТ 34.003.90 Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.
10. ГОСТ 34.320.96 Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы данных
11. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

12. ГОСТ 34.602-89 Информационная технология. Техническое задание на создание автоматизированной системы.
- 13.Эпов, А. А. Терминологический словарь-справочник по информатике / А. А. Эпов [и др.]. – Волгоград: ИУНЛ ВолгГТУ, 2013. – 80 с.
- 14.Глушаков, С.В. Базы данных: Учебный курс/ С.В. Глушаков, Д.В. Ломотько – Харьков: Фолио; Ростов н/Д: Феникс; Киев: Абрис, 2000. – 504 с.
- 15.Моргунов, Е. П. Язык SQL. Базовый курс: учеб.-практ. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова; Postgres Professional. — М., 2017. — 256 с. Тоже URL:
- 16.Коцюба, И. Ю. Основы проектирования информационных систем: Учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков.— СПб.: Ун-т ИТМО, 2015. Тоже URL: <https://books.ifmo.ru/file/pdf/1705.pdf>



## ПРИЛОЖЕНИЯ

- Таблицы сущностей информационной системы

### academicdegree

Комментарии к таблице: Список ученых степеней

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
AD_id ( <i>Первичный</i> )	int(11)	Нет			
AD_name	varchar(255)	Нет			ученая степень
AD_BrieflyName	varchar(255)	Нет			Краткое обозначение

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	AD_id	2	A	Нет	
AD_name_UNIQUE	BTREE	Да	Нет	AD_name	2	A	Нет	

### academicgroups

Комментарии к таблице: Академические группы

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
AG_Code ( <i>Первичный</i> )	varchar(45)	Нет			Код группы
AG_specialty	int(11)	Нет		specialtylist -> SL_id	Специальность
AG_YearOfStart	year(4)	Нет			Год начала
AG_YearOfEnd	year(4)	Нет			Код окончания
AG_NumCuorse	int(10)	Нет	1		Курс
AG_Curator	int(11)	Да	NULL	teacherprofile -> TP_UserID	Куратор
AG_Headman	int(11)	Да	NULL	studentprofile -> SP_id	Староста
AG_FormOfStudy	int(11)	Нет		formatofstudy -> FOS_id	Форма обучения

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null
PRIMARY	BTREE	Да	Нет	AG_Code	2	A	Нет
groupsCode_UNIQUE	BTREE	Да	Нет	AG_Code	2	A	Нет
AG_Headman_UNIQUE	BTREE	Да	Нет	AG_Headman	2	A	Да
SpecialtyID_idx	BTREE	Нет	Нет	AG_specialty	2	A	Нет
CuratorID_idx	BTREE	Нет	Нет	AG_Curator	2	A	Да
Headman_idx	BTREE	Нет	Нет	AG_Headman	2	A	Да
FormatOfStudy_idx	BTREE	Нет	Нет	AG_FormOfStudy	2	A	Нет

## academictitle

Комментарии к таблице: Ученое звание

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии
AT_id (Первичный)	int(11)	Нет			
AT_name	varchar(255)	Нет			Ученое звание
AT_BrieflyName	varchar(255)	Нет			Краткое обозначение

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null
PRIMARY	BTREE	Да	Нет	AT_id	4	A	Нет
AT_name_UNIQUE	BTREE	Да	Нет	AT_name	4	A	Нет

## attendancelesson

Комментарии к таблице: Посещаемость

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
AL_LessonInfo_id ( <i>Первичный</i> )	int(11)	Нет		lessoninfo -> LI_id	ИД занятия	
AL_Student_id ( <i>Первичный</i> )	int(11)	Нет		studentprofile -> SP_id	Студент	
AL_NumberHours	int(10)	Да	0		Количество астрономических, которые посетил студент	
AL_LessonGrande	int(11)	Да	0		Заработанные баллы	
AL_Coments	varchar(255)	Да	NULL		Коментарий	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	AL_LessonInfo_id	13	A	Нет	
				AL_Student_id	13	A	Нет	
FK_Student_id_idx	BTREE	Нет	Нет	AL_Student_id	13	A	Нет	

## departments

Комментарии к таблице: Кафедры

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
DEP_id ( <i>Первичный</i> )	int(11)	Нет				
DEP_Name	varchar(255)	Нет			Название кафедры	
DEP_Faculty_id	int(11)	Нет		faculties -> FCT_id	Факультет	
DEP_Head_id	int(11)	Да	NULL	teacherprofile -> TP_UserID	ЗавКаферы	
DEP_Abbreviation	varchar(255)	Нет			Аббревиатура	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	DEP_id	2	A	Нет	
DEP_Name_UNIQUE	BTREE	Да	Нет	DEP_Name	2	A	Нет	
Faculty_idx	BTREE	Нет	Нет	DEP_Faculty_id	2	A	Нет	

HeadTeacher_idx	BTREE	Нет	Нет	DEP_Head_id	2	A	Да	
-----------------	-------	-----	-----	-------------	---	---	----	--

## discipline

Комментарии к таблице: Таблица дисциплин

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
DISC_id ( <i>Первичный</i> )	int(11)	Нет			ИД дисциплины	
DISC_name	varchar(255)	Нет			Название дисциплины	
DISC_LeadTeacher_id	int(11)	Да	NULL	teacherprofile -> TP_UserID	Ведущий преподаватель	
DISC_Description	varchar(255)	Да	NULL		Описание дисциплины	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	DISC_id	3	A	Нет	
DISC_name_UNIQUE	BTREE	Да	Нет	DISC_name	3	A	Нет	
LeadTeacher_idx	BTREE	Нет	Нет	DISC_LeadTeacher_id	3	A	Да	

## disciplineformcontrol

Комментарии к таблице: Список дисциплин и форм кнтроля для них

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
DFC_Discipline_id ( <i>Первичный</i> )	int(11)	Нет		discipline -> DISC_id	ID дисциплины	
DFC_FormControl_id ( <i>Первичный</i> )	varchar(255)	Нет		formofcontrol -> FOC_Abbreviation	Форма контроля	
DFC_Teacher_id	int(11)	Нет		teacherprofile -> TP_UserID	Ответственный преподаватель	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	DFC_Discipline_id	1	A	Нет	
				DFC_FormControl_id	1	A	Нет	
Discipline_id_idx	BTREE	Нет	Нет	DFC_Discipline_id	1	A	Нет	
Teacher_id_idx	BTREE	Нет	Нет	DFC_Teacher_id	1	A	Нет	
FormOfControl_idx	BTREE	Нет	Нет	DFC_FormControl_id	1	A	Нет	

## faculties

Комментарии к таблице: Факультеты

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
FCT_id ( <i>Первичный</i> )	int(11)	Нет				
FCT_name	varchar(255)	Нет			Название факультета	
FCT_Deans	int(11)	Да	NULL	teacherprofile -> TP_UserID	Декан	
FCT_Abbreviation	varchar(255)	Нет			Аббревиатура	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	FCT_id	1	A	Нет	
FCT_name_UNIQUE	BTREE	Да	Нет	FCT_name	1	A	Нет	
TeacherDean_idx	BTREE	Нет	Нет	FCT_Deans	1	A	Да	

## formatofstudy

Комментарии к таблице: Форма обучения

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
FOS_id ( <i>Первичный</i> )	int(11)	Нет				
FOS_Name	varchar(255)	Нет			Форма обучения	

#### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	FOS_id	3	A	Нет	
FOS_Name_UNIQUE	BTREE	Да	Нет	FOS_Name	3	A	Нет	

## formofcontrol

Комментарии к таблице: Таблица форм контроля

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
FOC_Abbreviation ( <i>Первичный</i> )	varchar(255)	Нет				
FOC_Name	varchar(255)	Нет				

#### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	FOC_Abbreviation	5	A	Нет	
FOC_Abbreviation_UNIQUE	BTREE	Да	Нет	FOC_Abbreviation	5	A	Нет	
FOC_Name_UNIQUE	BTREE	Да	Нет	FOC_Name	5	A	Нет	

## lessoninfo

Комментарии к таблице: Информация о занятии

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
LI_id ( <i>Первичный</i> )	int(11)	Нет			ид занятия	
LI_Discipline_id	int(11)	Нет		discipline -> DISC_id	Дисциплина	
LI_Teacher	int(11)	Нет		teacherprofile -> TP_UserID	Преподаватель	
LI_date	date	Нет			Дата проведения занятия	
LI_AcademGroup_code	varchar(45)	Нет		academicgroups -> AG_Code	Академ. Группа	

LI_TypeLesson_id	varchar(65)	Нет		typelesson -> TL_id	Тип занятия	
LI_LessonTopic	varchar(255)	Нет			Тема занятия	
LI_LessonNumber_id	int(10)	Да	NULL	lessonnumber -> LN_Number	Номер пары	
LI_StartTime	timestamp	Да	NULL		Время начала занятия	
LI_EndTime	timestamp	Да	NULL		Время окончания занятия	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	LI_id	5	A	Нет	
LI_id_UNIQUE	BTREE	Да	Нет	LI_id	5	A	Нет	
FK_Discipline_id_idx	BTREE	Нет	Нет	LI_Discipline_id	5	A	Нет	
FK_TypeLesson_id_idx	BTREE	Нет	Нет	LI_TypeLesson_id	5	A	Нет	
FK_LessonNumer_idx	BTREE	Нет	Нет	LI_LessonNumber_id	5	A	Да	
FK_AcademicGroups_id_idx	BTREE	Нет	Нет	LI_AcademGroup_code	5	A	Нет	
FK_Teacher_id_idx	BTREE	Нет	Нет	LI_Teacher	5	A	Нет	

## lessonnumber

Комментарии к таблице: Расписание пар

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
LN_Number ( <i>Первичный</i> )	int(10)	Нет			Номер пары	
LN_StartTime ( <i>Первичный</i> )	time	Нет			Начало пары	
LN_EndTime ( <i>Первичный</i> )	time	Нет			Окончание пары	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	LN_Number	5	A	Нет	
				LN_StartTime	5	A	Нет	
				LN_EndTime	5	A	Нет	
LN_Number_UNIQUE	BTREE	Да	Нет	LN_Number	5	A	Нет	
LN_StartTime_UNIQUE	BTREE	Да	Нет	LN_StartTime	5	A	Нет	
LN_EndTime_UNIQUE	BTREE	Да	Нет	LN_EndTime	5	A	Нет	

## specialtylist

Комментарии к таблице: Список специальностей

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
SL_id ( <i>Первичный</i> )	int(11)	Нет				
SL_Code	varchar(255)	Нет			Код специальности	
SL_Name	varchar(255)	Нет			Название специальности	
SL_Profile_id	int(11)	Нет		specialtyprofile -> SProf_id	Профиль подготовки	
SL_Faculty_id	int(11)	Нет		faculties -> FCT_id	Факультет	
SL_Abbreviation	varchar(255)	Нет			Аббревиатура	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	SL_id	2	A	Нет	
SP_Code_UNIQUE	BTREE	Да	Нет	SL_Code	2	A	Нет	
SpecialtyProfiles_idx	BTREE	Нет	Нет	SL_Profile_id	2	A	Нет	
FacultyID_idx	BTREE	Нет	Нет	SL_Faculty_id	2	A	Нет	

## specialtyprofile

Комментарии к таблице: Таблица профилей(направленностей)

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
SProf_id ( <i>Первичный</i> )	int(11)	Нет				
SProf_name	varchar(255)	Нет			Название профиля подготовки	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	SProf_id	2	A	Нет	

## studentprofile

Комментарии к таблице: Студенты

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
SP_id ( <i>Первичный</i> )	int(11)	Нет		users -> User_id		



SP_Surname	varchar(255)	Нет			Фамилия	
SP_Name	varchar(255)	Нет			Имя	
SP_MiddleName	varchar(255)	Да	NULL		Отчество	
SP_BrieflyName	varchar(255)	Да	NULL		ФИО инициалы	
SP_Gender	enum('М', 'Ж')	Нет			Пол	
SP_DataOfBirth	date	Нет			Дата рождения	
SP_TypeOfStudy	enum('Б', 'К')	Нет			Тип обучения(бюджет/Коммерция)	
SP_NumberOfBook	varchar(255)	Да	NULL		Номер зачетной книжки	
SP_AcademGroup_id	varchar(255)	Нет		academicgroups -> AG_Code	Код группы	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	SP_id	6	A	Нет	
SP_NumberOfBook_UNIQUE	BTREE	Да	Нет	SP_NumberOfBook	6	A	Да	

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
AcademGroup_id_idx	BTREE	Нет	Нет	SP_AcademGroup_id	6	A	Нет	

## studyload

Комментарии к таблице: Учебная нагрузка дисциплины

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
SL_DISC_id <i>(Первичный)</i>	int(11)	Нет		discipline -> DISC_id	id дисциплины	
SL_TypeLesson_code <i>(Первичный)</i>	varchar(65)	Нет		typelesson -> TL_id	Вид занятия	
SL_NumberHours	int(10)	Нет			Количество часов	
SL_Teacher_id <i>(Первичный)</i>	int(11)	Нет		teacherprofile -> TP_UserID	Преподаватель	
SL_AcademGroup_code <i>(Первичный)</i>	varchar(65)	Нет		academicgroups -> AG_Code	Академическая группа	
SL_AdditionalLoad	tinyint(4)	Да	0		Доп. Нагрузка	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	SL_DISC_id	4	A	Нет	
				SL_TypeLesson_code	4	A	Нет	
				SL_Teacher_id	4	A	Нет	
				SL_AcademGroup_code	4	A	Нет	
DISC_id_idx	BTREE	Нет	Нет	SL_DISC_id	4	A	Нет	
SL_AcademGroup_idx	BTREE	Нет	Нет	SL_AcademGroup_code	4	A	Нет	
Teacher_idx	BTREE	Нет	Нет	SL_Teacher_id	4	A	Нет	
TypeLesson_idx	BTREE	Нет	Нет	SL_TypeLesson_code	4	A	Нет	

## subgroups

Комментарии к таблице: Подгруппы

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
SG_AcademicGroup_id <i>(Первичный)</i>	varchar(65)	Нет		academicgroups -> AG_Code	Код группы	
SG_SubGroups <i>(Первичный)</i>	int(11)	Нет			Номер подгруппы	
SG_Student <i>(Первичный)</i>	int(11)	Нет		studentprofile -> SP_id	Студент	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	SG_AcademicGroup_id	2	A	Нет	
				SG_SubGroups	2	A	Нет	
				SG_Student	2	A	Нет	
FK_Student_idx	BTREE	Нет	Нет	SG_Student	2	A	Нет	

## teacherprofile

Комментарии к таблице: Профиль преподавателя

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
TP_UserID ( <i>Первичный</i> )	int(11)	Нет		users -> User_id	Ссылка но провиль в таблице пользователей	
TP_Surname	varchar(255)	Нет			Фамилия	
TP_Name	varchar(255)	Нет			Имя	
TP_MiddleName	varchar(255)	Да	NULL		Отчество	

TP_BrieflyName	varchar(255)	Да	NULL		ФИО инициалы	
TP_Gender	enum('М', 'Ж')	Нет			Пол	
TP_DataOfBirth	date	Нет			Дата Рождения	
TP_Degree	int(11)	Да	NULL	academicdegree -> AD_id	Ученая степень	
TP_AcademicTitle	int(11)	Да	NULL	academictitle -> AT_id	Ученое звание	
TP_Department	int(11)	Да	NULL	departments -> DEP_id	Кафедра	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	TP_UserID	6	A	Нет	
TP_login_UNIQUE	BTREE	Да	Нет	TP_UserID	6	A	Нет	
AcademicDegree_idx	BTREE	Нет	Нет	TP_Degree	6	A	Да	
AcademicTitle_idx	BTREE	Нет	Нет	TP_AcademicTitle	6	A	Да	
Departament_idx	BTREE	Нет	Нет	TP_Department	6	A	Да	

## typelesson

Комментарии к таблице: типы занятий

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
TL_id <i>(Первичный)</i>	varchar(65)	Нет			Краткое обозначение	
TL_Name	varchar(255)	Нет			Полное название	

## Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	TL_id	3	A	Нет	
TL_id_UNIQUE	BTREE	Да	Нет	TL_id	3	A	Нет	
TL_Name_UNIQUE	BTREE	Да	Нет	TL_Name	3	A	Нет	

## users

Комментарии к таблице: Пользователи

Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME
User_id <i>(Первичный)</i>	int(11)	Нет				

User_login	varchar(255)	Нет			Логин	
User_email	varchar(255)	Нет			e-mail	
User_Passwd	varchar(255)	Нет			Пароль	
User_Role	int(10)	Нет		usersrole -> UR_id	Роль пользователя	

### Индексы

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	User_id	12	A	Нет	
User_email_UNIQUE	BTREE	Да	Нет	User_email	12	A	Нет	
User_login_UNIQUE	BTREE	Да	Нет	User_login	12	A	Нет	
Role_idx	BTREE	Нет	Нет	User_Role	6	A	Нет	

### usersrole

Комментарии к таблице: Таблица ролей пользователей

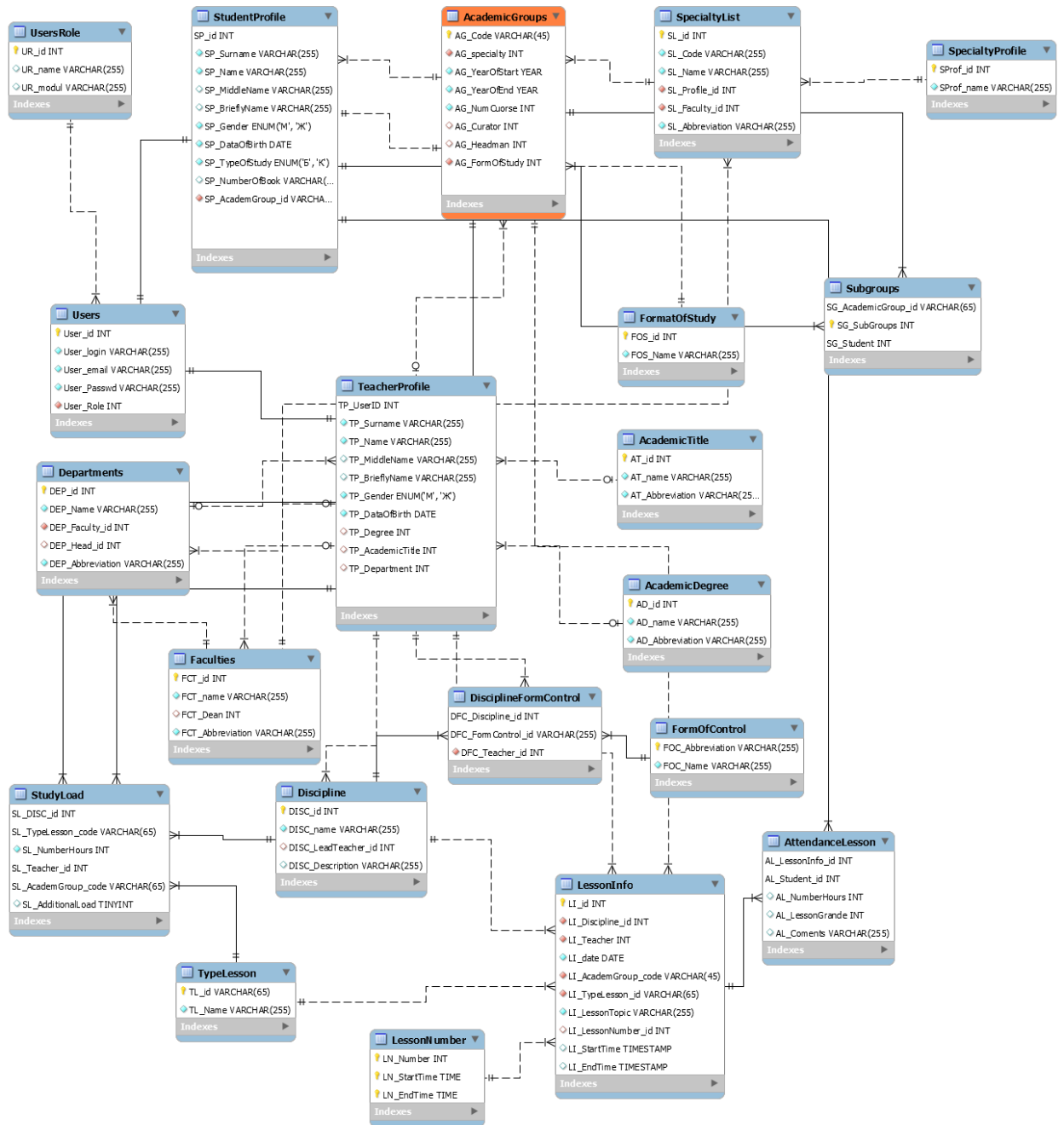
Столбец	Тип	Null	По умолчанию	Ссылки на	Комментарии	MIME

UR_id ( <i>Первичный</i> )	int(10)	Нет			ИД Роли
UR_name	varchar(255)	Да	<i>NULL</i>		Название Роли
UR_modul	varchar(255)	Да	<i>NULL</i>		Модуль доступа

**Индексы**

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null
PRIMARY	BTREE	Да	Нет	UR_id	3	A	Нет
UR_name_UNIQUE	BTREE	Да	Нет	UR_name	3	A	Да

- Полная EER-диаграмма



- Основной функционал системы

```
if($db==NULL) return NULL;
```

```
$query = "Select User_id as ID, u.UR_modul As Modules from users join
usersrole u on u.UR_id = users.User_Role
where User_login = :login And User_Passwd = :password";
```

```
$stmt = $db->prepare($query);
```

```
$stmt->execute(array('login'=>$login, 'password'=>$password));
```

```
$result_query = $stmt->fetch();
```

```
if(!$result_query)
```

```
    return false;
```

```
$UserID = $result_query['ID'];
```

```
$UserModules = $result_query['Modules'];
```

```
$User_key = uniqid();
```

```
if(isset($_COOKIE['jagu_user_key'])) {
```

```
    unset($_COOKIE['jagu_user_key']);
```

```
}
```

```
setcookie('jagu_user_key', $User_key);
```

```
$_SESSION[$User_key] = Array("UserID"=>$UserID,
'UserModules'=>$UserModules);
```



```

Header("Location: /Modules/" . $UserModules);

return TRUE;
}

// Проверка: авторизован-ли пользователь, если да, то вернуть его роль
function isAuthUser()
{
    if($_SESSION[$_COOKIE['jagu_user_key']])
        return $_SESSION[$_COOKIE['jagu_user_key']]['UserModules'];
    else
        return NULL;
}

// Отмена авторизации пользователя
function logout()
{
    if($_SESSION[$_COOKIE['jagu_user_key']]){
        unset($_SESSION[$_COOKIE['jagu_user_key']]);
        session_destroy();
        unset($_COOKIE['jagu_user_key']);
    }
}

// Перенаправление пользователя на модуль, соответствующий его роли
function RedirectUser()

```

```
{  
    if(isAuthUser() != null)  
        header('Location: /Modules/'.isAuthUser());  
}
```

- Конфигурация global.php

```
<?php  
// Конфигурация сайта  
const SITE_NAME='JAGU';  
const SITE_DESCRIPTION = "Журнал академической группы вуза";  
const SITE_URL ='www.jagu.ru';  
$Page_Title = SITE_NAME; //Глобальная переменная - title page
```

```
// Конфигурация БД
```

```
const DB_HOST='localhost';  
const DB_USERNAME = 'Admin';  
const DB_PASSWORD = 'admin';  
const DB_NAME = 'jagu';
```

- Открытый репозиторий GitHub

<https://github.com/MedvedevKP/jagu.git>