

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**РАЗРАБОТКА ПОДСИСТЕМЫ ФОРМИРОВАНИЯ РЕЙТИНГА  
ЭФФЕКТИВНОСТИ РАБОТЫ ПРЕПОДАВАТЕЛЕЙ, АСПИРАНТОВ И  
МАГИСТРАНТОВ ДЛЯ СИСТЕМЫ «ИНФОБЕЛГУ: УЧЕБНЫЙ  
ПРОЦЕСС»**

Выпускная квалификационная работа  
обучающегося по направлению подготовки  
09.03.02 Информационные системы и технологии  
очной формы обучения, группы 07001408  
Боброва Владислава Николаевича

Научный руководитель  
к.т.н., доцент  
Штифанов А.И.

БЕЛГОРОД 2018

## РЕФЕРАТ

Разработка подсистемы формирования рейтинга эффективности работы преподавателей, аспирантов и магистрантов для системы «ИнфоБелГУ: Учебный процесс» – Бобров Владислав Николаевич, выпускная квалификационная работа бакалавра, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 66, включая приложения 95, количество рисунков 55, количество таблиц 4, количество используемых источников 21.

**КЛЮЧЕВЫЕ СЛОВА:** рейтинг эффективности работы, модуль, повышение качества работы, информация, данные.

**ОБЪЕКТ ИССЛЕДОВАНИЯ:** система рейтинга эффективности работы для НИУ «БелГУ».

**ПРЕДМЕТОМ ИССЛЕДОВАНИЯ:** методы и средства оценивания качества работы НИУ «БелГУ».

**ЦЕЛЬ РАБОТЫ:** создание подсистемы рейтингования различных категорий учащихся, ППС и сотрудников научных подразделений для автоматизированной системы управления учебным процессом «ИнфоБелГУ: Учебный процесс».

**ЗАДАЧИ ИССЛЕДОВАНИЯ:** изучить теоретические данные предметной области, сделать её анализ и на основании полученной информации спроектировать подсистему рейтинга эффективности работы, для объективной оценки магистрантов, аспирантов и преподавателей и их стимулирование в виде материальных благ.

**ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ:** в результате работы была спроектирована и программно реализована подсистема рейтинга эффективности работы.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Аналитическая часть.....	7
1.1 Краткая характеристика НИУ «БелГУ».....	7
1.2 Принципы организации рейтинга эффективности работы.....	9
1.3 Анализ существующих разработок в области рейтингования.....	11
1.2 Постановка задачи.....	16
2 Проектирование информационной системы формирования рейтинга эффективности работы преподавателей, аспирантов и магистрантов для системы «ИнфоБелГУ: Учебный процесс».....	19
2.1 Информационное обеспечение комплекса задач.....	19
3. Программная реализация и тестирование.....	31
3.1 Особенности разработки рейтинга эффективности работы.....	31
3.2 Разработка модулей рейтинга эффективности работы.....	37
3.3 Тестирование.....	43
ЗАКЛЮЧЕНИЕ.....	63
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	65
ПРИЛОЖЕНИЕ А.....	67
ПРИЛОЖЕНИЕ Б.....	79
ПРИЛОЖЕНИЕ В.....	86
ПРИЛОЖЕНИЕ Г.....	93

## ВВЕДЕНИЕ

Проблема оценки качества профессиональной деятельности отдельного педагога и педагогического коллектива в целом во все времена являлась одной из актуальных и в то же время труднейших проблем, касающихся взаимоотношений не только внутри самого профессионально-педагогического сообщества, но и взаимодействия образовательных систем с социумом. Являясь по сути своей социальным институтом, система образования всегда находится в центре пристального внимания общества, которое хочет знать, насколько эффективно последняя реализует предписанные ей социальные функции.

Актуальность разработки системы рейтингования обусловлена тем, что рейтинг как система оценки по формальным показателям обладает неоспоримыми достоинствами – ясностью критериев, простотой реализации и невозможностью произвольно завышать, или занижать оценку в зависимости от того, чья именно деятельность оценивается.

Магистратура, аспирантура являются основными формами подготовки научно-педагогических и научных кадров, предоставляющими гражданам РФ возможность повышения уровня образования, научной и педагогической квалификации.

Рейтингование магистрантов и аспирантов также является актуальной задачей и способствует выявлению и поддержке талантливых, активно работающих перспективных кадров, стимулированию научной деятельности молодых ученых и информированию научной общественности о достижениях магистрантов и аспирантов.

Система оценки качества профессиональной деятельности различных категорий учащихся, профессорско-преподавательского состава (ППС), факультетов и кафедр вуза должна быть по возможности проста и понятна не только руководителям образовательных учреждений и самим участникам рейтинга, но и другим заинтересованным лицам, а в ее основе должны лежать в первую очередь очевидные и общепринятые показатели.

Получение результатов рейтинговой оценки, позволит объективно оценить работу магистрантов, аспирантов, преподавателей и научных работников, определить узкие места и недоработки по каким-либо направлениям деятельности и, с учетом этого, правильно спланировать работу на будущий год. Такая оценка, несомненно, положительно отражается на стремлении сотрудников к повышению своей научной квалификации, профессионализма, росту педагогического мастерства, а это, в свою очередь, будет способствовать повышению качества подготовки специалистов и росту рейтинга университета. Показатели, используемые при рейтинговой оценке, ориентируют профессорско-преподавательский состав университета на решение основных задач развития, как подразделений университета, так и всего университета в целом.

Оценка рейтингования позволяет принимать оптимальные управленческие решения, связанные с укреплением кадрового потенциала, аттестацией преподавателей, их профессиональной мотивацией, определением приоритетных целей в области управления качеством подготовки специалистов в вузе.

Целью выпускной квалификационной работы создание подсистемы рейтингования различных категорий учащихся, ППС и сотрудников научных подразделений для автоматизированной системы управления учебным процессом «ИнфоБелГУ: Учебный процесс».

Задачами данной выпускной квалификационной работы являются:

- анализ предметной области;
- формирование требований к программному обеспечению;
- проектирование подсистемы рейтингования;
- программная реализация веб-приложения;
- испытание программного обеспечения и анализ результатов.

Разработка и внедрение информационной системы рейтингования позволит упорядочить и автоматизировать процессы накопления, обработки и предоставления информации об успешности научной деятельности различных

категорий учащихся, ППС и сотрудников научных подразделений.

В первой главе производится анализ предметной области, связанный с функционированием Белгородского государственного национального исследовательского университета и принципами организации рейтинга эффективности работы. Также будет произведён обзор существующих разработок других учебных заведений.

Во второй главе будет описан процесс проектирования подсистемы рейтингования эффективности работы.

В третьей главе будет описан процесс разработки подсистемы рейтингования эффективности работы, а также его апробация.

## 1 Аналитическая часть

### 1.1 Краткая характеристика НИУ «БелГУ»

Белгородский государственный национальный исследовательский университет – это научно-образовательный комплекс, обеспечивающий качественную подготовку кадров широкому спектру медицинских, социальных, гуманитарных, экономических, управленческих, филологических, физико-математических и информационных специальностей в центральной части России. С 1996 года является государственным университетом, а в 2010 получил статус «Национальный исследовательский университет». На текущий момент ректором университета является профессор, доктор политических наук Полухин Олег Николаевич. Юридический адрес: Россия, Белгород, индекс – 308015, улица Победы 85. Сайт университета расположен по адресу: <https://www.bsu.edu.ru/bsu/>. [1]

НИУ «БелГУ» состоит из 7 институтов и 3 факультетах, не входящих в состав институтов, 2 колледжа, Старооскольский филиал; 105 кафедр, 24 из которых базовые; более 600 предприятий, являющимися базами практик; около 1100 докторов и кандидатов наук, 14 академиков и членов-корреспондентов наук. [2]

Структура учебных единиц НИУ «БелГУ»:

- а) институт юридический;
- б) институт педагогический:
  - 1) факультет физической культуры;
  - 2) факультет дошкольного, начального и специального образования;
  - 3) факультет историко-филологический;
  - 4) факультет математики и естественнонаучного образования;
  - 5) факультет иностранных языков;
  - 6) факультет психологии.
- в) институт медицинский;

- 1) медицинский колледж;
- 2) центр дополнительного медицинского и фармацевтического образования, аккредитации и сертификации.
- г) институт межкультурной коммуникации и международных отношений:

- 1) факультет подготовительный.
- 2) институт управления:
- 3) высшая школа управления.
- д) институт экономики;
- е) институт инженерных технологий и естественных наук;
- ж) факультет горного дела и природопользования;
- з) факультет журналистики;
- и) факультет социально-теологический;
- к) инжиниринговый колледж;
- л) старооскольский филиал.

В рамках концепции непрерывного образования осуществляет обучение по образовательным программам высшего образования и среднего профессионального образования, дополнительным профессиональным программам и дополнительным общеобразовательным программам, основным профессиональным программам послевузовского медицинского и фармацевтического образования в интернатуре. Образовательная деятельность в вузе ведется по 180 направлений подготовки бакалавриата, магистратуры и специалитета; около 300 дополнительных профессиональных программ повышения квалификации и переподготовки.

Общее количество обучающихся – около 25,8 тысяч человек. Из них 23 тысячи всех регионов России и 2,8 тысячи иностранных студентов из 91 страны мира. Каждый год выпускаются больше 5 тысяч высококвалифицированных специалистов разных областей. [2]

## 1.2 Принципы организации рейтинга эффективности работы

Цель выпускной квалификационной работы – программная реализация рейтинга эффективности работы магистрантов, аспирантов, преподавателей в системе «ИнфоБелГУ: Учебный процесс». Результатом выполнения работы будет являться повышение эффективности и оперативности работы сотрудников НИУ «БелГУ».

Основными задачами рейтингования преподавателей, аспирантов и магистрантов являются:

- обеспечение объективности оценки результатов деятельности и определение текущего состояния качества деятельности научных подразделений, лабораторий, кафедр, центров, факультетов и институтов;
- формирование информационного ресурса, отражающего деятельность институтов, факультетов, кафедр, преподавателей и сотрудников, как по отдельности, так и во взаимодействии друг с другом;
- создание условий для положительной динамики роста квалификации и профессионализма сотрудников, развития творческой инициативы НПП, магистрантов и аспирантов;
- получение единых комплексных критериев для оценки и контроля уровня эффективности факультетов, научных подразделений, кафедр, НПП, магистрантов и аспирантов;
- стимулирование видов деятельности, ориентированных на повышение рейтинга университета и качества его деятельности;
- формирование системы материального и морального стимулирования.

Принципы организации рейтинга эффективности работы:

- критерии рейтинговой оценки результатов деятельности НПП, магистров, аспирантов, лабораторий, центров, кафедр, факультетов и институтов максимально соответствуют критериям, необходимыми для определения типа и вида образовательного учреждения высшего образования, критериям общероссийской системы оценки эффективности деятельности

образовательных учреждений высшего образования, а также критериям эффективности реализации Программы развития Белгородского государственного университета.

- объективность и полнота предоставляемой информации;
- открытость, оперативность и систематичность рейтингования;
- использование одновременно нескольких форм рейтинговой оценки;
- компетентность экспертов, проводящих оценивание;
- стимулирование нпр, магистрантов, аспирантов, лабораторий, центров, кафедр, факультетов и институтов к улучшению деятельности;
- принцип наследования: показатели структурных подразделений учитывают показатели структурных подразделений предыдущего уровня.

Рейтинг эффективности работы деятельности НПр, магистрантов, аспирантов, лабораторий, центров, кафедр, факультетов и институтов НИУ «БелГУ» позволит:

- оценить соответствие профессиональной компетентности сотрудника должностным квалификационным требованиям, а также его личный вклад в решение задач кафедры, факультета, института;
- оценить качество учебно-научной деятельности магистрантов и аспирантов;
- определить сотрудников, занимающих лидирующие позиции в отдельных видах деятельности (в учебно-методической, научной и инновационной, организационно-воспитательной работе);
- стимулировать повышение активности преподавателей в учебно-методической, научной и инновационной деятельности, организационно-воспитательной работе;
- упростить процесс выдвижения кандидатур на замещение научно-педагогических должностей;
- перераспределять научно-педагогический потенциал кафедры, факультета, института, университета при изменении приоритетов решаемых задач.

### – ы1.3 Анализ существующих разработок в области рейтингования

На сегодняшний день в России в практике современной высшей школы представлено более 40 методик измерения и оценки качества деятельности преподавателей. [3] Наиболее распространенным является расчет индивидуального рейтинга преподавателя и абсолютного рейтинга преподавателя. В практике российских ВУЗов можно выделить несколько подходов к решению оценки качества деятельности ППС: учет мнения студентов о качестве преподавания и личностных качествах преподавателя, рейтинговая оценка деятельности преподавателя, учет результатов обучения, оценка педагогического мастерства преподавателя.

К сожалению, поиск в сети Интернет готовых информационных или автоматизированных систем в области оценки качества преподавателей и сотрудников не увенчался успехом. Удалось найти только несколько информационных систем рейтингования студентов и одну для рейтингования ППС. По-видимому, информационные системы рейтингования ППС особо не рекламируются и находятся в закрытом доступе. Или же их просто не существует.

#### 1.3.1 Информационная система рейтингования студентов НИУ Томского политехнического университета

Данная информационная система реализована при помощи программного продукта «1С: Предприятие 8.2». [4]

В рейтинговой системе оценки студентов установлен перечень критериев, таких как: оценка, прохождение итогового контроля (дисциплина, сданная в срок либо с опозданием), форма итогового контроля, статус научного мероприятия, статус спортивного мероприятия, статус культурно- творческого мероприятия, призовое место, статус деятельности студента (староста, профорг и т.д.).

Для функционирования любой программы необходимо создать ряд объектов информационной системы. В данном случае это справочники, документы, перечисления, отчеты, регистры сведений и накоплений. Справочники в системе:

- а) справочник «физическое лицо» – предназначен для хранения индивидуальной информации;
- б) справочник «кафедры» – предназначен для хранения списка кафедр вуза;
- в) справочник «специальности» – предназначен для хранения списка специальностей вуза;
- г) справочник «группы» – предназначен для списка групп студентов обучающихся в вузе;
- д) справочник «дисциплины» – предназначен для хранения списка дисциплин, преподаваемых в вузе;
- е) справочник «конференции» – предназначен для хранения списка конференций, проводившихся в вузе;
- ж) справочник «спортивные соревнования» – предназначен для хранения списка соревнований, проводившихся в вузе.

Отчета «Рейтингование» представлен на рисунке 1.1.

Рейтингование - Рейтингование в вузе (1С:Предприятие)

Вариант отчета: Основной

Сформировать Настройка...

Группа: 17821 (2012 г.)

Начало периода: Начало этого года

Конец периода: Начало следующего полугодия

Параметры: Группа: 17821 (2012 г.)

Группа	Успеваемость	Научная деятельность	Соревнования	Творчество	Социальный статус	Итоговый балл	
№ ФЛ п/п							
17821 (2012 г.)							
1	Иванов Валентин Иванович	55	13	6	8	3	85
2	Глызин Максим Егорович	55	8	8			71
3	Берлиан Анатолий Павлович	50			10		60
4	Быков Леонид Андреевич	45					45

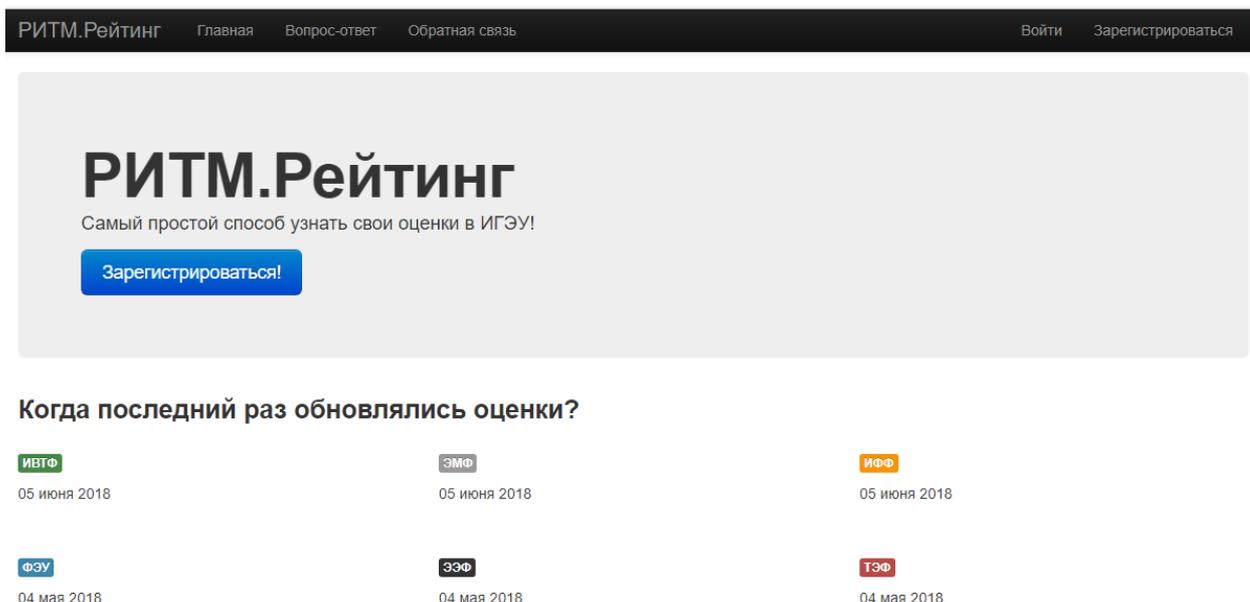
Рисунке 1.1 – Отчет «Рейтигование»

Отчет в системе Отчет «Рейтингование» – позволяет на основании выбранной группы и проведенным документам «Учебная успеваемость», «Научная деятельность», «Соревнования», «Творчество» и «Социальный статус» сформировать отчет «Рейтингование».

### 1.3.2 Система учёта рейтинга студентов в Ивановском государственном энергетическом университете

Используется автоматизированная система оценки [5]. Методика оценки. Комбинированный вариант оценки качества деятельности ППС представляет собой компиляцию ряда известных методов. На среднем уровне интеграции формируются таблицы для каждого преподавателя и оцениваются пять аспектов качества деятельности каждого сотрудника вуза. Первый аспект – относительное качество по результатам ежегодного отчета преподавателя оценивается от 0 до 5 баллов. В качестве эталона для сравнения заведующий кафедрой выбирает показатели лучшего преподавателя вуза в соответствующей номинации по итогам предыдущего года. Материалы для анализа направляются в Центр менеджмента качества образования (ЦМКО) ежегодно в виде отчета о победителях конкурса «Лучший преподаватель вуза» в соответствующей номинации («Руководство научной работой студентов», «Руководство научной работой аспирантов и докторантов», «Вклад в стратегию развития вуза», «Презентация научных работ», «Финансируемые проекты», «Вклад в улучшение качества процессов на кафедре или в вузе», «Награды» и другие индикаторы качества). Второй аспект – интегральное мнение коллег о профессиональных качествах преподавателя. Третий аспект – интегральное мнение студентов о качестве преподаваемого курса. Четвертый – самооценка готовности преподавателя к работе в системе стратегического управления качеством образования. И наконец, пятый – оценка по анкетным данным. Цель системы измерения, передачи и анализа данных о качестве работы ППС – устранение субъективизма процессов измерения и повышение уровня

автоматизации управления вузом. На рисунке 1.2 представлена система рейтингования «РИТМ.Рейтинг»



Рисунке 1.2. – Стартовая страница системы оценки в Ивановском государственном энергетическом университете

Данная система рейтингования охватывает только малую часть категорий рейтингования.

### 1.3.3 Информационная система рейтинговой оценки деятельности преподавателя в Курском государственном медицинском университете

Разработанный в КГМУ программный продукт «Рейтинг преподавателя Вуза» расположен по адресу <http://rating.kursksmu.net/> и имеет два пользовательских интерфейса: непосредственно пользовательский и интерфейс суперпользователя.[6] Первый является общедоступным, ко второму доступ имеет лишь администратор, который может отслеживать результаты рейтингования, что недоступно простому пользователю.

Ресурс <http://rating.kursksmu.net> является корпоративным и доступен только зарегистрированным пользователям в режиме входа из «Удаленного рабочего стола». После авторизации пользователь попадает на главную

страницу, где находятся вкладки: «Профиль», «Мои записи», «Мой рейтинг», «Инструкции». В разделе Инструкции можно найти полную информацию с пошаговыми иллюстрациями о том, как вносить данные и сохранять записи.

На рисунке 1.3 представлен интерфейс профиля системы рейтинга.

Рисунке 1.3 – Интерфейс страницы профиля преподавателя.

Для начала работы необходима регистрация, которую можно пройти при нажатии на вкладку «Профиль», где пользователь должен заполнить следующие строки: название кафедры, должность, фамилия, имя, отчество. Кроме этого при регистрации необходимо указать средневзвешенный импакт-фактор журналов, в которых были опубликованы статьи (сведения из eLIBRARY на момент заполнения показателей рейтинга), число цитирований автора их последних публикаций за последние 5 лет (сведения из eLIBRARY на момент заполнения показателей рейтинга), индекс Хирша (сведения из eLIBRARY на момент заполнения показателей рейтинга), импакт-фактор журналов, в которых были опубликованы статьи за отчетный период.

## 1.2 Постановка задачи

С 2013 года в Белгородском государственном национальном исследовательском университете функционирует автоматизированная система управления учебным процессом, планирования и организации проведения учебной деятельности подразделениями университета «ИнфоБелГУ: Учебный процесс».[7] Основана на модульной динамической учебной среде «Moodle».

В систему «ИнфоБелГУ: Учебный процесс» входят следующие основные подсистемы:

- «ИнфоБелГУ: Учебный процесс/Расписание» – подсистема составления расписаний занятий;
- «ИнфоБелГУ: Учебный процесс/Учебные планы/Нагрузка» – подсистема составления учебных рабочих планов и нагрузки преподавателей;
- «ИнфоБелГУ: Учебный процесс/Аудиторный фонд» – подсистема учёта занятости аудиторного фонда;
- «ИнфоБелГУ: Учебный процесс/Рейтингование» – подсистема общеуниверситетского мониторинга результативности научной и учебно-методической работы кафедр, научных подразделений и центров;
- «ИнфоБелГУ: Учебный процесс/Успеваемость» – подсистема учёта, мониторинга успеваемости обучающихся;
- «ИнфоБелГУ: Учебный процесс/Дипломы» – подсистема формирования дипломов для обучающихся;
- «ИнфоБелГУ: Учебный процесс/Личный кабинет студента» – подсистема просмотра своей успеваемости, посещаемости, заполнение электронного портфолио;
- «ИнфоБелГУ: Учебный процесс/Личный кабинет аспиранта» – подсистема заполнения портфолио;
- «ИнфоБелГУ: Учебный процесс/Диссертационные советы» – подсистема информационной поддержки диссертационных советов;

– «ИнфоБелГУ: Учебный процесс/Основные образовательные программы» – подсистема аккумуляции рабочих программ, учебных планов, практик и ГИА.

Одна из подсистем предназначена для проведения общеуниверситетского мониторинга результативности научной и учебно-методической работы кафедр, научных подразделений и центров. Именно она используется с 2014 года для осуществления рейтинговой оценки деятельности ППС и сотрудников научных подразделений. К сожалению, в ней отсутствует возможность добавления новых категорий и конструктор критериев рейтингования. Также со стороны пользователей было много нареканий на пользовательский интерфейс данной подсистемы.

Подсистема формирования рейтинга эффективности работы магистрантов, аспирантов, преподавателей СПО, ППС, сотрудников научных подразделений, кафедр, факультетов и институтов для системы «ИнфоБелГУ: Учебный процесс».

Обозначение – «ИнфоБелГУ: Учебный процесс/Рейтинг эффективности работы»

Назначение подсистемы «ИнфоБелГУ: Учебный процесс/Рейтинг эффективности работы»

Система предназначена для

- формирования рейтинга эффективности работы магистрантов, аспирантов, преподавателей СПО, ППС, сотрудников научных подразделений, кафедр, факультетов и институтов;
- создание и редактирование критериев рейтингования с помощью встроенного конструктора;
- хранения введенных данных и подгружаемых документов;
- построения отчетов о результатах рейтингования.

При проектировании подсистемы должны приниматься во внимание следующие требования:

- а) подсистема должна реализовывать собственное назначение;

б) подсистема должна быть интегрирована в портал «инфобелгу: учебный процесс» и автоматизировать основные функции для построения рейтинга эффективности работы;

в) структура базы данных подсистемы «инфобелгу: учебный процесс/рейтинг эффективности работы» должна расширять и дополнять структуру основной базы данных и соответствовать формату данных СУБД mysql;

г) пристальное внимание при разработке подсистемы должно быть уделено вопросам информационной безопасности, связанными в первую очередь с разграничениями прав доступа;

д) разрабатываемая подсистема должна включать в себя модули, разработанные на языке программирования php;

е) система должна иметь интуитивно ясный интерфейс и не должна требовать от пользователей специальной подготовки, не связанной с их профессиональными обязанностями.

#### Выводы по первому разделу

В данном разделе было рассмотрено описание предметной области, сформулированы задачи рейтингования эффективности работы, так же был проведен сравнительный анализ существующих разработок в области рейтингования. При анализе систем рейтингования на просторах сети-интернет очень мало было найдено систем рейтингования, что говорит, а том что данный вид систем либо же находится во внутренней сети учреждения к которой нету доступа, либо же информационные системы не популярны. Были поставлены технические требования к системе.

## 2 Проектирование информационной системы формирования рейтинга эффективности работы преподавателей, аспирантов и магистрантов для системы «ИнфоБелГУ: Учебный процесс»

### 2.1 Информационное обеспечение комплекса задач

Информационное обеспечение – это один из важнейших элементов информационной системы, оно предназначено для отображения информации, характеризующей состояние управляемого объекта. Является базовым критерием для принятия управленческих решений.

Информационное обеспечение в каждой информационной системе базируется на понятии информационной базы, под которой понимается совокупность упорядоченной информации, используемой при функционировании информационной системы, а также взаимосвязь различных частей этой системы. При этом всем совокупность упорядоченной информации должна соответствовать по содержанию и составу требованиям технических задач, которые решаются на её основе.

Информационная база – это сущность, которая влияет на эффективность всей системы, на возможности решения различных задач, решаемых данной информационной системой. В её состав входят:

- массивы, содержащие накапливаемые данные за определенный промежуток времени;
- массивы постоянной нормативно-справочной информации;
- массивы, содержащие текущие данные о состоянии управляемого объекта;
- массивы, содержащие данные, поступающие из внешней среды.

Основная задача организации информационного обеспечения состоит в адекватном отображении объекта управления и обеспечении информационных потребностей функциональных задач управления, а в данном случае,

потребностей задачи разработки рейтинга формирование эффективности работы преподавателей, аспирантов и магистрантов.

### 2.1.1 Информационная модель и ее описание

В соответствии с методологией IDEF0, которая может быть использована для моделирования широкого класса систем, построим диаграммы разрабатываемого рейтинга эффективности работы преподавателей, магистрантов и аспирантов. На рисунке 2.1 показана контекстная диаграмма модуля рейтинг эффективности работы.

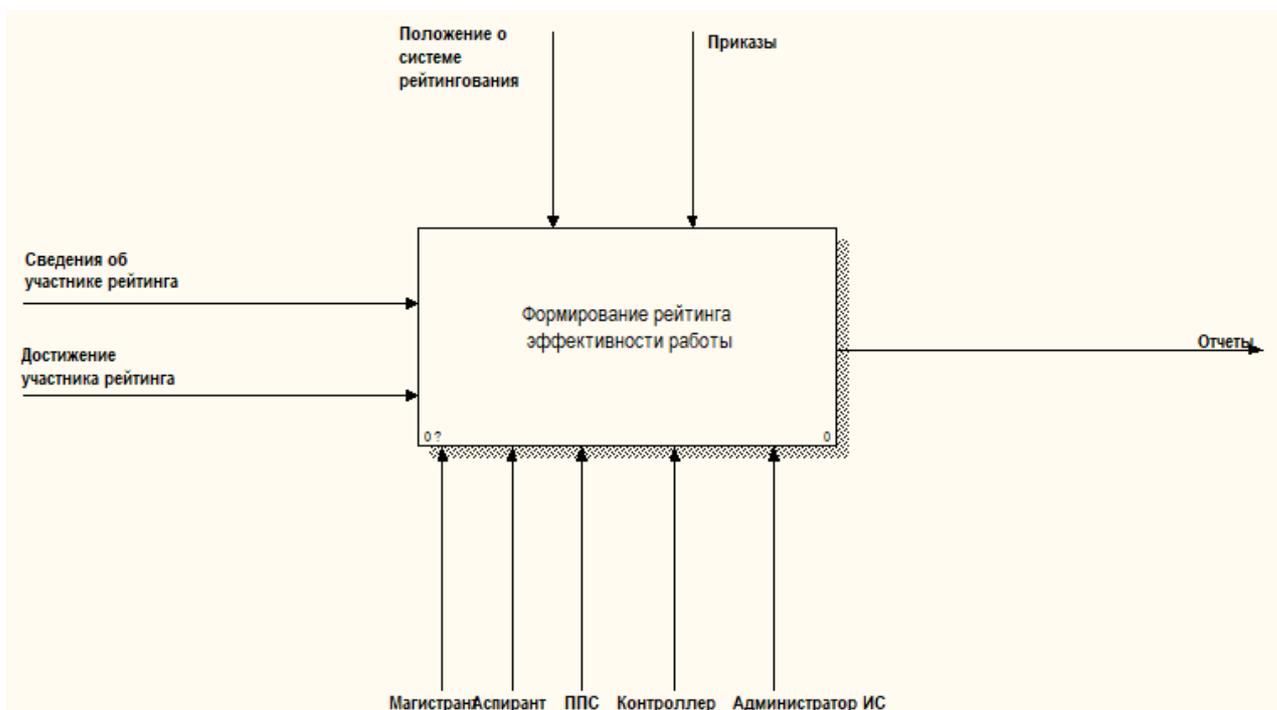


Рисунок 2.1 – Формирование рейтинга эффективности работы

Из контекстной диаграммы видно, что входными данными проектируемого модуля являются сведения об участнике рейтинга и достижения участника рейтинга.

Выходными данными проектируемого модуля являются отчеты, содержащие в себе множество отсортированных данных по различным параметрам.

В качестве управления используется положение о системе рейтингования, приказы, а в качестве механизмов, необходимых для выполнения работы выступают магистрант, аспирант, ППС, контроллер, администратор ИС.

После описания модуля рейтинга эффективности работы проводится его декомпозиция на более малые функциональные узлы – процессы.

При декомпозиции контекстной диаграммы модуля можно выделить ряд основных функций выполняемым данным модулем:

- просмотр и редактирование работ в портфолио;
- добавление работы в портфолио;
- проверка работы в портфолио;
- формирование отчетов участников рейтинга.

Детализация контекстной диаграммы представлена на рисунке 2.2.

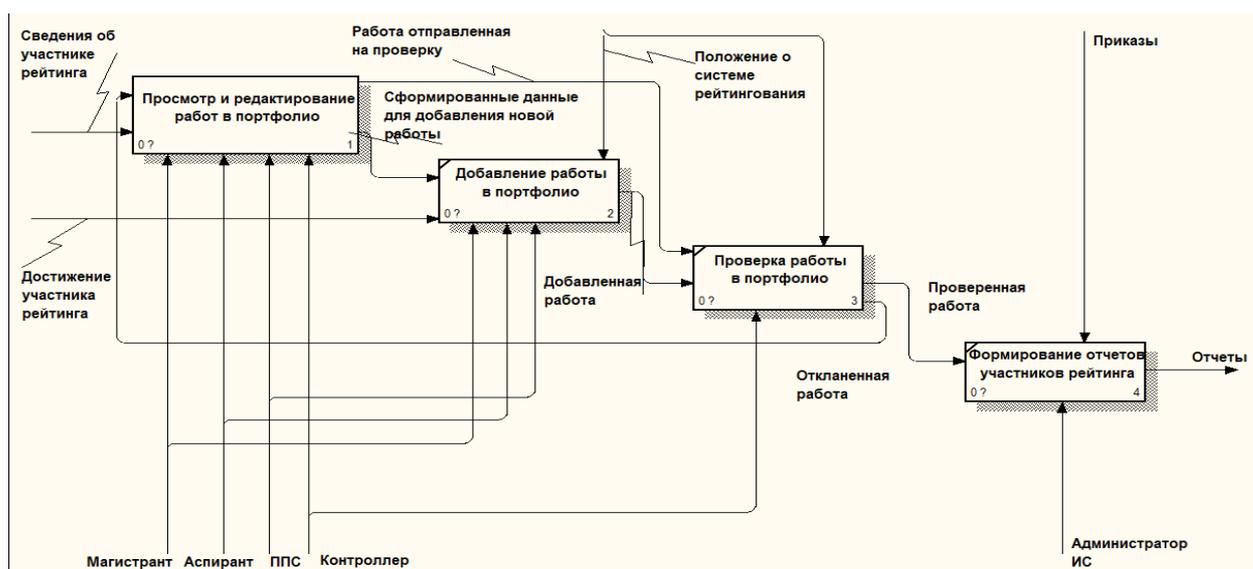


Рисунок 2.2 – Декомпозиция рейтинга эффективности работы

Процесс «Просмотр и редактирование работ в портфолио» будет детализирован при помощи методологии IDEF3.

Методология IDEF3 является стандартом документирования технологических процессов, позволяющая схематически отображать последовательность изменений свойств процессов и объектов, так называемые сценарии.

На рисунке 2.3 представлен сценарий работы функции «Просмотр и редактирование работ в портфолио».

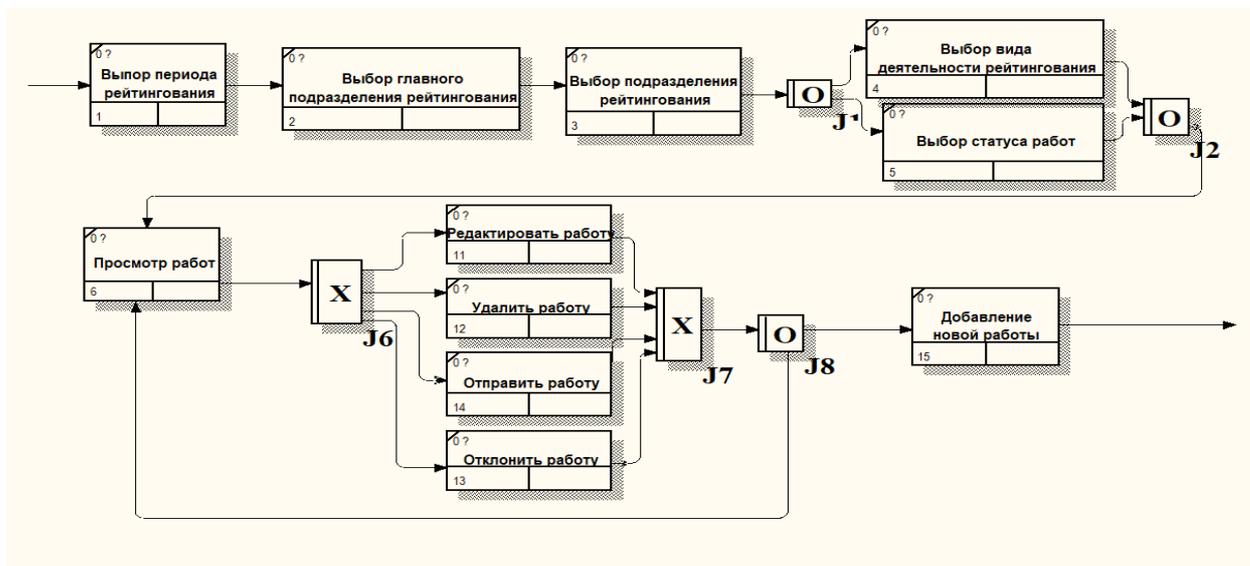


Рисунок 2.3 – Декомпозиция рейтинга эффективности работы

Декомпозиция «Просмотра и редактирования» показывает процесс, происходящий при просмотре и редактировании работ в портфолио рейтинга эффективности работы. Данный процесс почти является линейным за исключением некоторых событий:

- редактирование работ в портфолио;
- удаление работы в портфолио;
- отправление работы в портфолио;
- отклонение работы в портфолио.

### 2.1.2 Проектирование логической модели базы данных

Логическая модель включает описания объектов БД и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных. Логическая модель создается без привязки к способам физического хранения объектов БД. Таким образом, основное назначение логической модели БД является описание предметной

области будущей БД и выделение в ней основных объектов и признаков, по которым экземпляры этих объектов можно различать.

На основе анализа предметной области рейтинга эффективности работы магистрантов, аспирантов и ППС была спроектирована логическая схема базы данных, удовлетворяющая всем требованиям ИС. Были выделены сущности проектируемой базы данных и спроектирована логическая модель схемы данных. На рисунке 2.4 представлена концепция логической схемы базы данных.

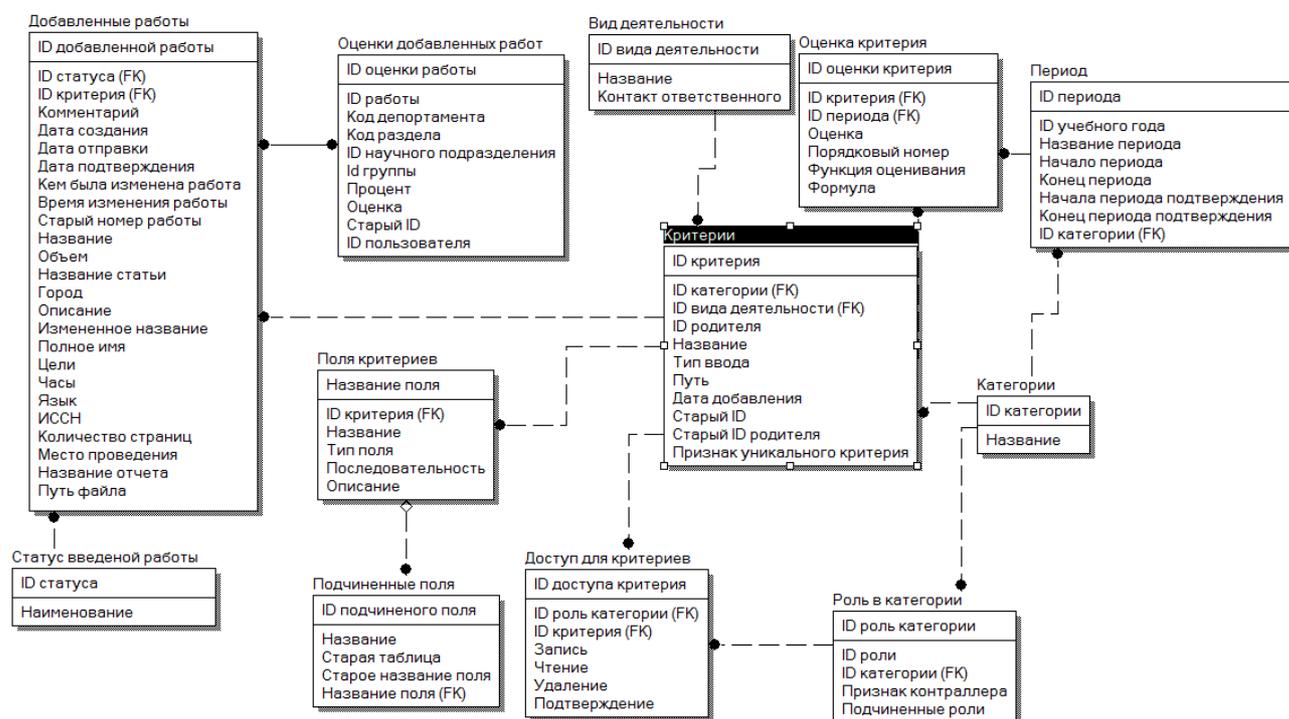


Рисунок 2.4 – Логическая схема базы данных рейтинга эффективности работы

Данная логическая схема базы данных состоит из 12 сущностей:

- «Добавленные работы» – предназначена для хранения информации о добавленных работах в портфолио рейтинга эффективности работы;
- «Статус введенной работы» – предназначена для хранения статусов введенных работ (добавлено, отправлено, отклонено, подтверждено);
- «Оценки добавленных работ» – хранит в себе оценки об работах;
- «Вид деятельности» – предназначена для хранения видов деятельности;

- «Оценка критерия» – предназначена для хранения оценок критериев;
- «Период» – предназначена для хранения периодов рейтингования;
- «Категории» – предназначена для хранения категорий рейтингования;
- «Критерии» – хранит в себе перечень всех доступных критериев;
- «Поля критериев» – хранит в себе поля для критериев, которые отображаются пользователю при добавлении или редактировании работы;
- «Подчиненные поля» – предназначена для хранения подчиненных полей, данные поля необходимы для формирования выпадающих списков;
- «Доступ для критериев» – предназначена для хранения прав доступа критериев;
- «Роль в категории» – данная сущность, хранящая в себе информацию о том какие роли, относятся к каким категориям.

### 2.1.3 Построение физической модели базы данных

Физическая модель БД строится на основе логической. Ее основным назначением является определение физического размещения данных, методов доступа и техники индексирования и будущих объемов БД. Таким образом, под физическим проектированием понимается определение количества и размеров файлов данных, определение табличных пространств, методик хранения и доступа к данным БД.

На рисунке 2.5 представлена физической схемы данных рейтинга эффективности работы магистрантов, аспирантов и преподавателей.

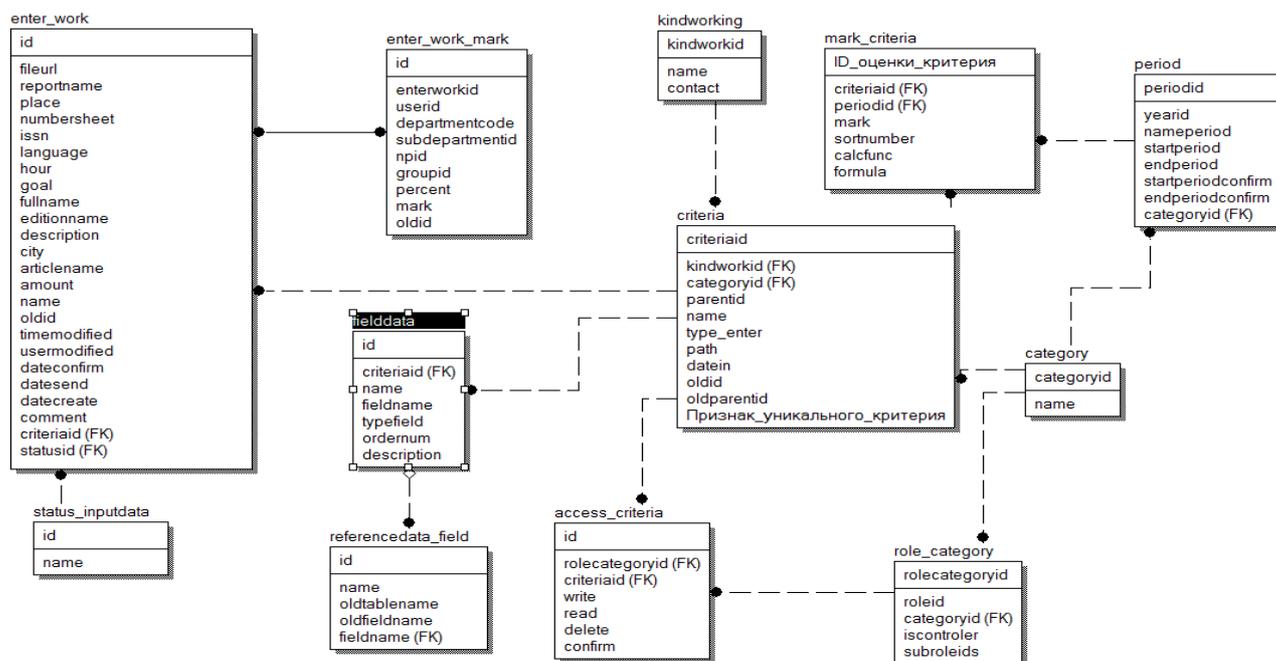


Рисунок 2.5 – Концепция физической схемы базы данных рейтинга эффективности работы

Данная схема базы данных была спроектирована таким образом, что она позволяет:

- добавлять новые критерии рейтингования эффективности работы;
- добавлять новые категории рейтингования эффективности работы;
- хранить историю периодов рейтингования эффективности работы;
- разграничивать доступ к критериям рейтингования эффективности работы без ущерба существующей структуре базы данных.

Были созданы первичные и вторичные ключи для каждой таблицы базы данных.

Первичный ключ необходим для того, что сохранить ограниченность целостности данных, он должен быть уникальным.

Вторичный ключ предназначен для ограничения целостности связей между таблицами, подчиненная таблица не может ссылаться на несуществующие записи главной таблицы. Это необходимо что бы исключить ошибки при дальнейшем использовании базы данных.

В физической схеме данных были опущены следующие префиксы таблиц:

- префикс «mdl» – обозначает то, что данная таблица относится к системе Moodle;
- префикс «bsu» – обозначает то, что данная таблица относится к системе «ИнфоБелГУ: Учебный процесс»;
- префикс «rating» – обозначает то, что данная таблица относится к подсистеме рейтинга эффективности работы.

Это было сделано для того, чтобы упростить наглядное представление физической схемы базы данных.

В реальной базе данных к каждой из таблиц рейтинга эффективности работы был добавлен составной префикс «mdl\_bsu\_rating». Все скрипты для создания схемы базы данных представлены в приложении А.

На рисунке 2.6 представлена информация о таблице «enter\_work».

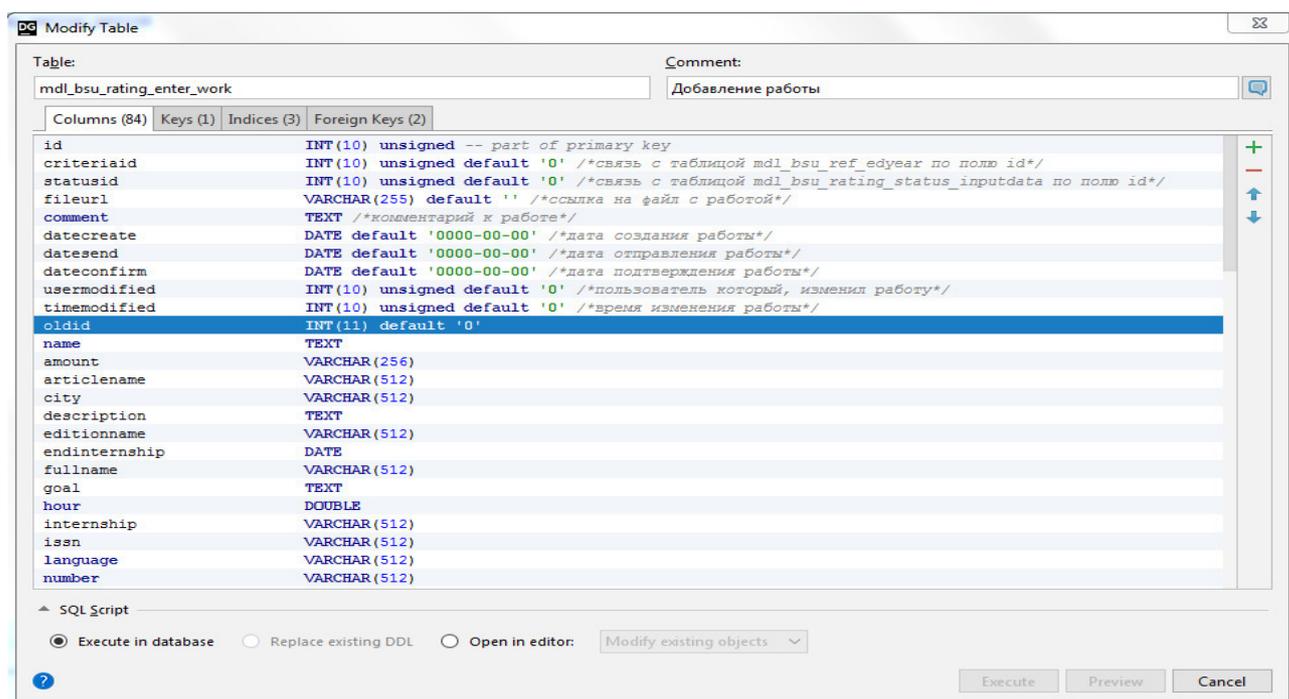


Рисунок 2.6 – Таблица «enter\_work»

Данная таблица хранит в себе информацию о работах. Каждая работа относится к определенному критерию, а для каждого критерия существуют перечень полей необходимых для заполнения. Из-за этого таблица «enter\_work»

при необходимости будет пополняться новыми атрибутами. В таблице 2.1 представлено часть полей таблицы «enter\_work».

Таблица 2.1. – Обозначение полей таблицы «enter\_work»

id	Идентификатор работы
criteriaid	Идентификатор критерия
statusid	Индетификатор статуса
fileurl	Путь файла
comment	Комментарий к работе
datecreate	Дата создания работы
datesend	Дата отправки работы
dateconfirm	Дата подтверждения работы
usermodified	Пользователь, который изменил работу
timemodified	Время изменения работы
oldid	Устаревший идентификатор работы
name	Название работы
amount	Объем работы
articlename	Название статьи
city	Город
description	Описание
editionname	Название издания
endentership	Дата стажировки
issn	ИССН
language	язык
number	номер

Все оценки работ и процент вклада в работу, а так же множество других полей необходимых для работы «рейтинга эффективности работы» хранится в таблице «enter\_work\_mark» представленной на рисунке 2.7.

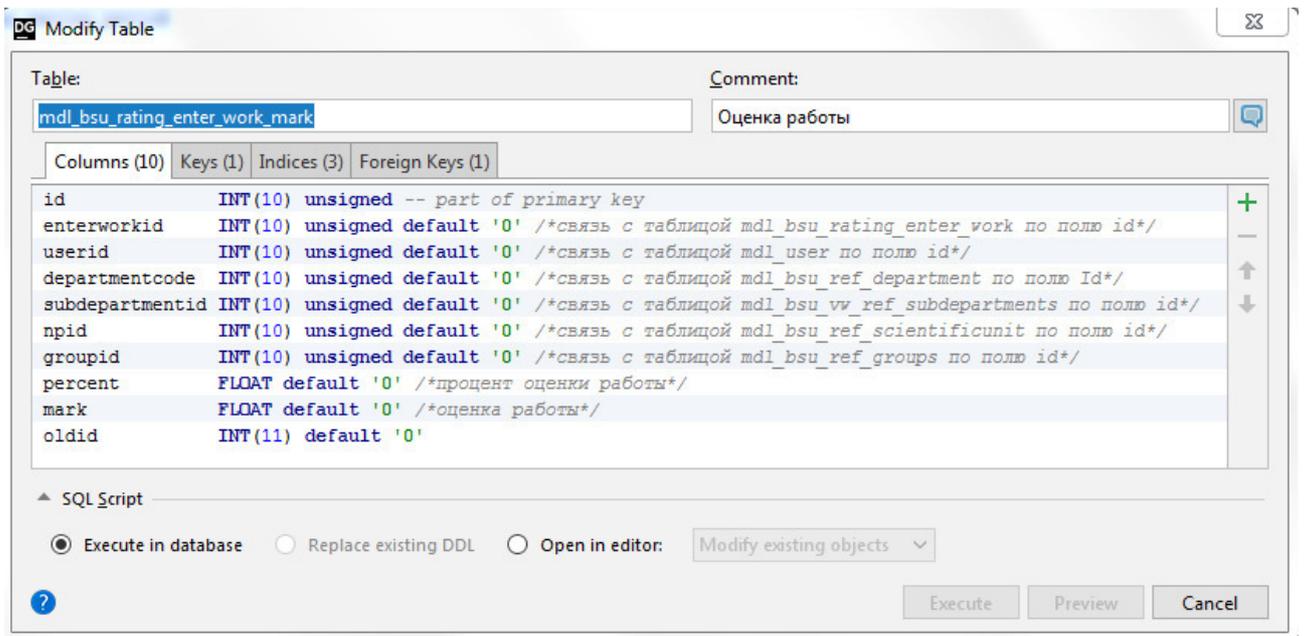


Рисунок 2.7 – Таблица «enter\_work\_mark»

Данная таблица содержит ряд полей, представленных в таблице 2.2.

Таблица 2.2 – Обозначение полей таблицы «enter\_work»

id	Идентификатор оценки работы
enterworkid	Идентификатор работы
userid	Идентификатор пользователя
departmentcode	Идентификатор главного подразделения
subdepartmentcode	Идентификатор подразделения
npid	Номер научного подразделения
groupid	Номер группы
percent	Процент вклада в работу
mark	Оценка
oldid	Старый идентификатор

Все доступные критерии хранятся в таблице «criteria». Данная таблицы представлена на рисунке 2.8.

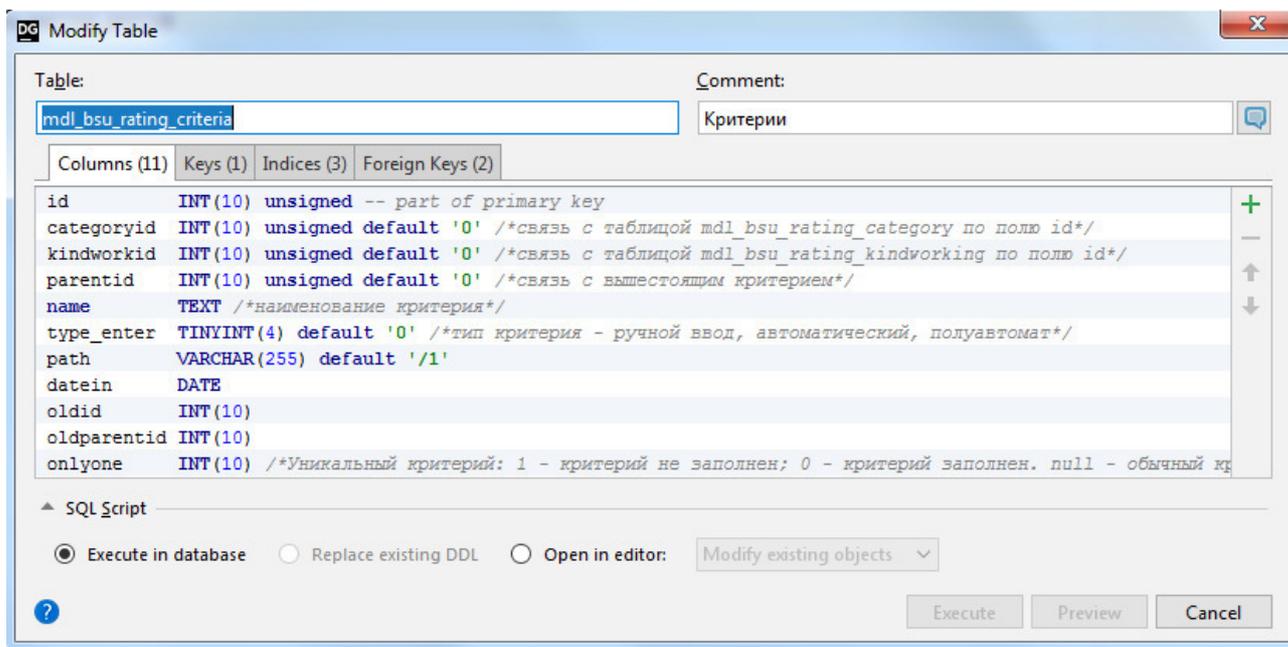


Рисунок 2.8 – Таблица «criteria»

Поля, содержащиеся в таблице «criteria» представлены в таблице 2.3.

Таблица 2.3 – Обозначение полей таблицы «criteria»

id	Идентификатор критерия
categoryid	Идентификатор категории
kindworkid	Идентификатор вида деятельности
parentid	Идентификатор родителя
name	Название
type_enter	Тип критерия
path	Путь
datein	Дата добавления критерия
oldid	Устаревший идентификатор
oldparentid	Устаревший идентификатор родителя
onlyone	Признак уникальности критерия

Весы критериев или так называемые оценки критериев хранятся в таблице «mark\_criteria» представленной на рисунке 2.9.

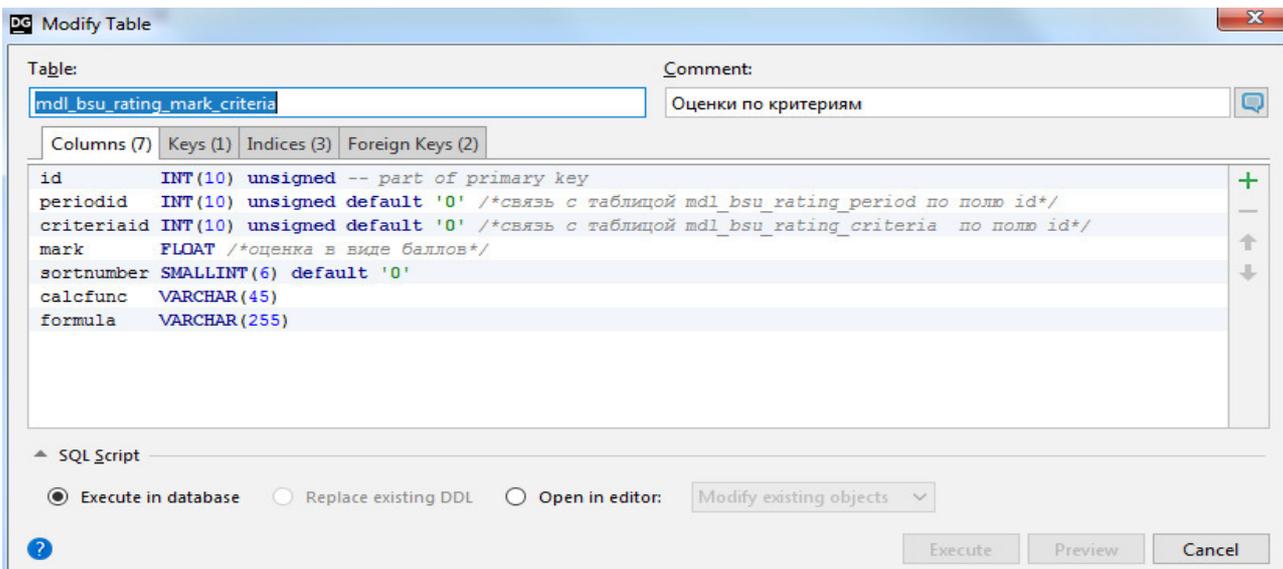


Рисунок 2.9 – Таблица «mark\_criteria»

Поля, содержащиеся в таблице «mark\_criteria» представлены в таблице 2.4.

Таблица 2.4 – Обозначение полей таблицы «mark\_criteria»

id	Идентификатор оценки критерия
periodid	Идентификатор периода
criteriaid	Идентификатор критерия
mark	Оценка критерия
sortnumber	Порядковый номер
calcfunc	Функция для подсчета баллов критерия
formula	Формула

### Выводы по второму разделу

В данном разделе было произведено проектирование и моделирование модуля. Это включает в себя информационное обеспечение, информационную контекстную диаграмму «Формирования рейтинга эффективности работы», логическую и физическую схемы базы данных. На основании контекстной диаграммы и получении знаний на преддипломной практике была спроектирована база данных системы рейтингования, которая является основой всей системы.

### 3. Программная реализация и тестирование

#### 3.1 Особенности разработки рейтинга эффективности работы

##### 3.1.1 Шаблон проектирования

Был выбран шаблон проектирования MVC (Model View Controller) для подсистемы рейтинга эффективности работы. Данный шаблон проектирования подразумевает под собой разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента:

- а) модель;
- б) представление;
- в) контроллер.

Модель включает в себя бизнес-логику приложения, SQL запросы, включает методы выборки, обработки и предоставления конкретных данных. Она никогда не взаимодействует с пользователем, модель общается исключительно с контроллером. Так же модель не должна генерировать HTML или же любой другой код отображения. Код отображения должен обрабатываться в видах.

Представление отображает данные о модели пользователю, формирует ответ в виде html/css от сервера и передаёт его браузеру пользователя.

Контроллер – это связующее звено между моделью, видом и другими компонентами приложения. Он отвечает за обработку запросов пользователя.

Данное разбиение системы на 3 компонента, позволяет разграничивать логику работы программы таким образом, что модификация каждого компонента может осуществляться независимо. Благодаря этому, наращивании функционала подсистемы не составит особого труда.

Компонент подразумевает под собой некую часть кода. Каждая из которых играет одну из ролей представления, контроллера и модели. Каждый

компонент, это отдельный файл с расширением `php`, он же является объектом класса.

На рисунке 3.1 представлен принцип работы компонента «контроллер».

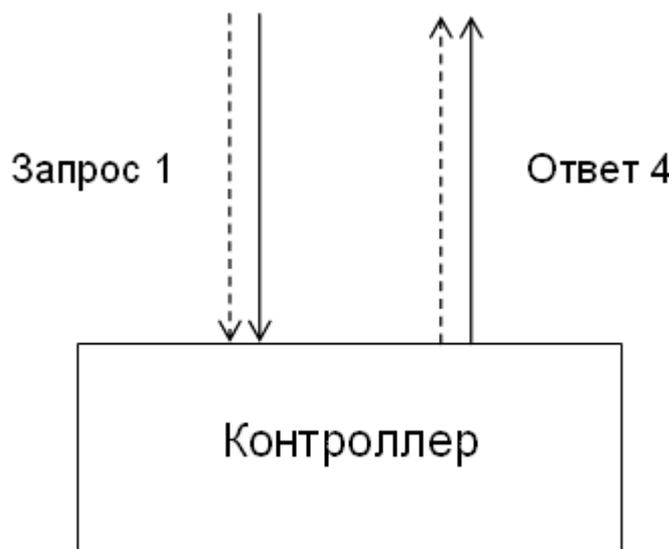


Рисунок 3.1 – Принцип работы контроллера

Как только контроллер получит запрос 1, он его проанализирует, и в зависимости от того какой запрос пришел, контроллер выдаст уже заранее подготовленные ответы на запросы. Варианты ответов на запрос:

а) отправит сообщение об ошибке, если страница не найдена, то контроллер может отдать только `http`-заголовок с кодом ошибки;

б) если такая страница существует и запрос от пользователя корректен, то, в зависимости от того, какой запрос пришел на просмотр или модификацию данных, контроллер сделает обращение к соответствующему методу модели.

Концептуальная схема шаблона MVC представлена на рисунке 3.2.

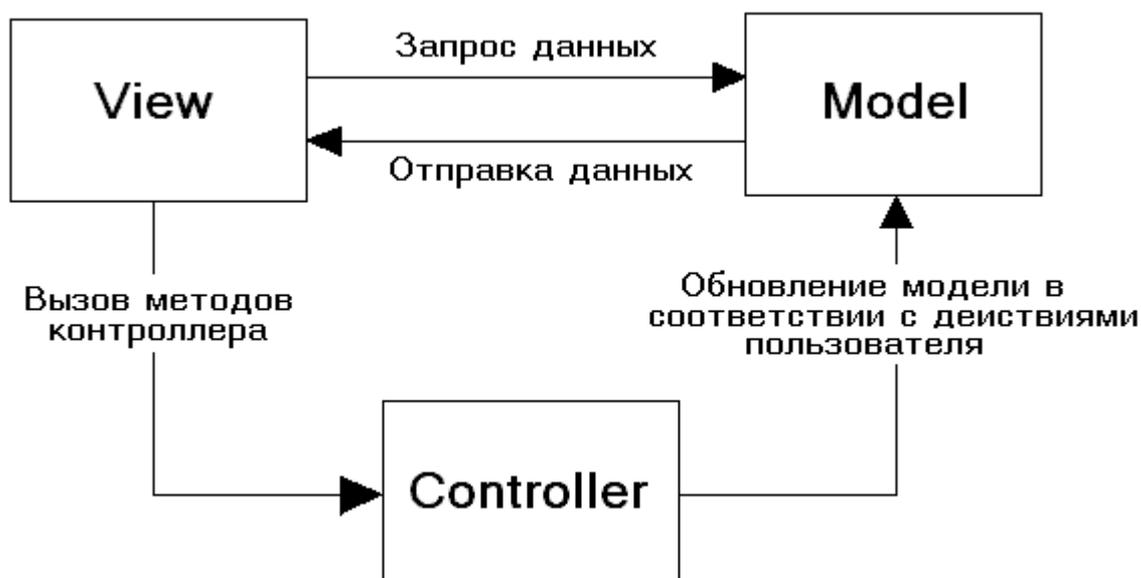


Рисунок 3.2 – Концептуальная схема шаблона MVC

Например мы рассмотрим работу портфолио рейтинга эффективности работы. После того как пользователь попал на страницу портфолио, затем запрос передаётся контроллеру, при этом запускается метод обрабатывающий этот запрос и подгружающий нужную модель, модель в свою очередь отправит данные, которые нужно отобразить пользователю.

### 3.1.2 Инструментальные средства разработки

Разработка программного обеспечения осуществляется с использованием различных специализированных программ, которые упрощают процесс разработки программного продукта.

Системе «ИнфоБелГУ: Учебный процесс» – это интернет ресурс, работающий на стеке технологий PHP 5.6, Apache и MySQL. В связи с этим фактором были выбраны следующие программные продукты для разработки подсистемы рейтинга эффективности работы магистрантов, аспирантов и преподавателей:

- PhpStorm 2017.3.4;
- DataGrip 2017.2.3;

– MySQL Workbench 6.3.

На рисунке 3.3 представлен интерфейс программы PhpStorm.

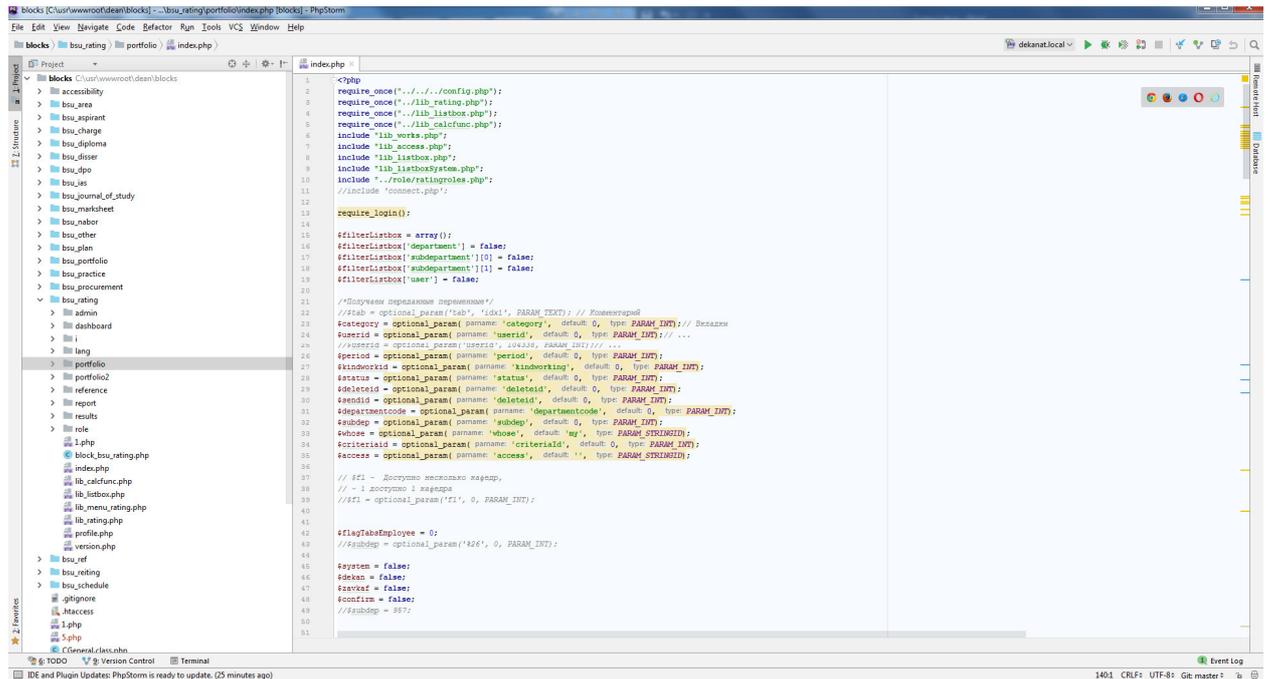


Рисунок 3.3 – Интерфейс программы PhpStorm

PhpStorm 2017.3.4 – это интегрированная среда разработки на языке программирования PHP с интеллектуальным редактором, которая предотвращает синтаксические ошибки на лету, изменение, автодополнение кода. Поддерживающий передовые веб технологии:

- HTML5;
- CSS;
- Sass;
- Less;
- SCSS;
- Stylus;
- Compass;
- CoffeeScript;
- TypeScript;
- Jade;
- Emmet;

– JavaScript.

На рисунке 3.4 представлен интерфейс программы DataGrip.

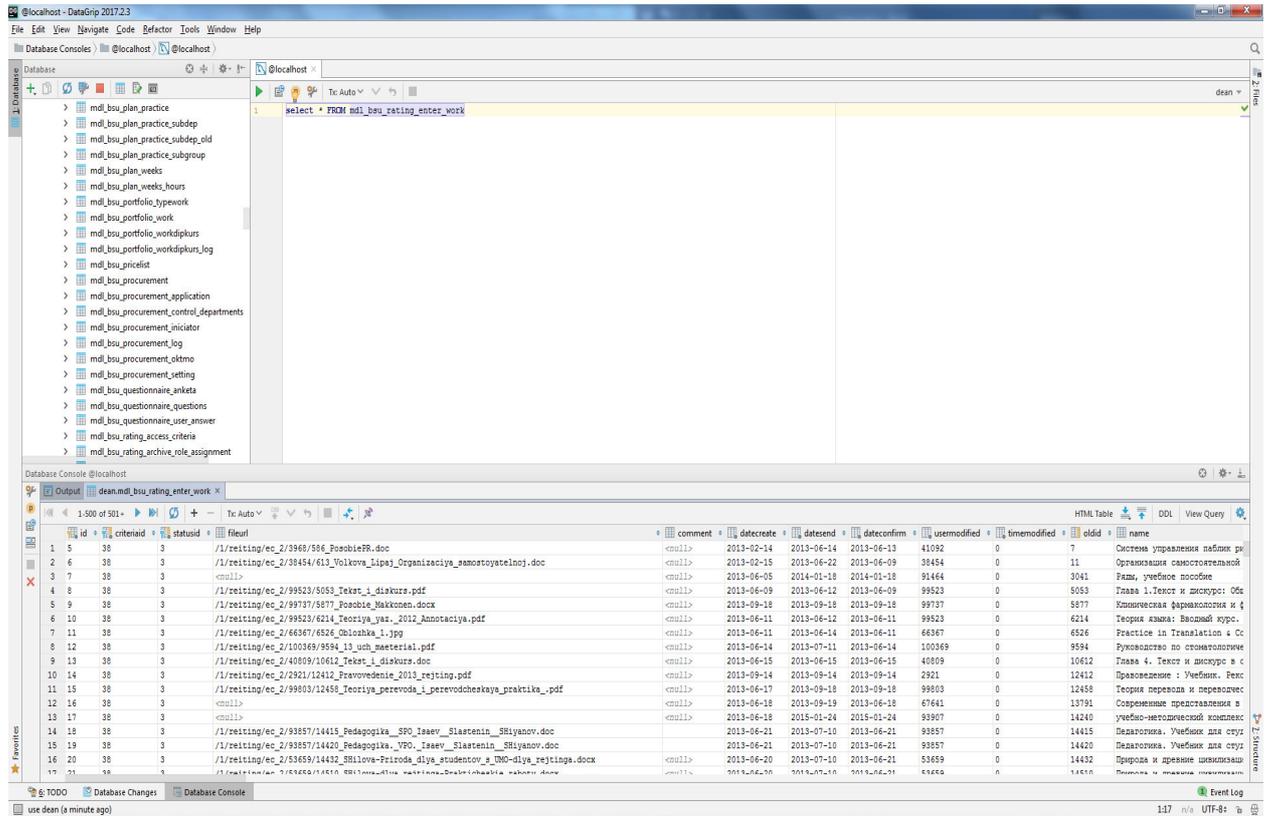


Рисунок 3.4 – Интерфейс программы DataGrip

DataGrip 2017.2.3 – это система управления базами данных, содержащая консоль в себе интеллектуальную консоль запросов SQL. Данная СУБД позволяет выполнять запросы, импортировать, экспортировать данные из базы данных, формировать отчеты в виде HTML, так же проводить анализ скриптов в режиме реального времени и предлагать возможные варианты исправления. Это все позволяет ускорить разработку программного продукта.

MySQL Workbench 6.3 – это система управления базами данных, основная особенность состоит в том что она свободно распространяется под лицензией GNU GPL. Одна из основных особенностей MySQL Workbench – это возможность схематического построения базы данных. Визуальное представление базы данных даёт полное понимание предметной области.

Во-вторых, в программе существует встроенный SQL-редактор позволяющий строить SQL запросы любой сложности.



– слабо развит конструктор кодов.

## 3.2 Разработка модулей рейтинга эффективности работы

### 3.2.1 Модуль портфолио

По шаблону проектирования MVC был разработан модуль «Портфолио» рейтинга эффективности работы.

На рисунке 3.6 представлена структура файлов модуля портфолио рейтинга эффективности работы.

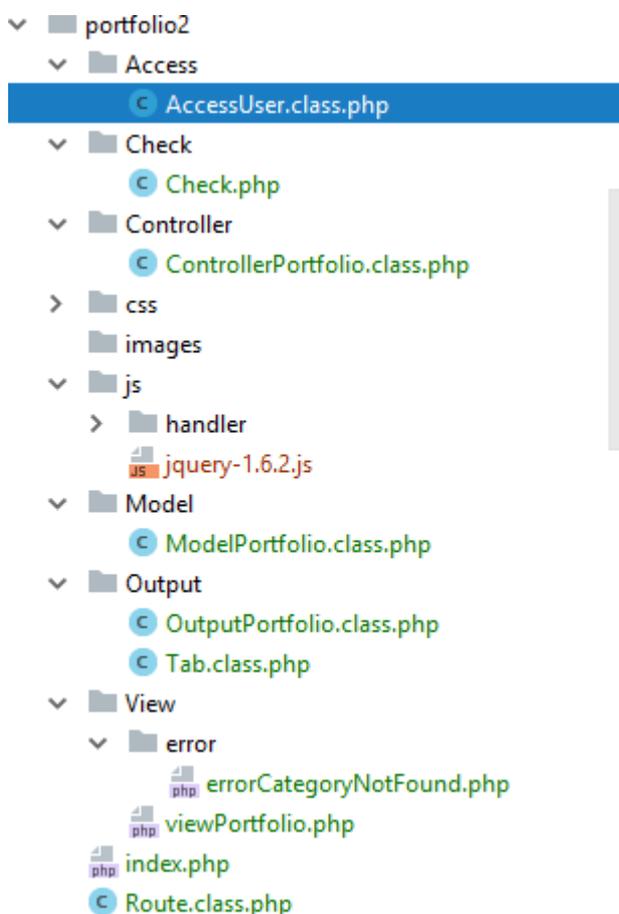


Рисунок 3.6 – Структура файлов модуля портфолио рейтинга эффективности работы

Вход в программу начинается с файла `index.php`. Это происходит после того, как пользователь перейдет по ссылке ведущей на портфолио рейтинга эффективности работы. На рисунке 3.7 представлена структура файла `index.php`.

```

<?php
require_once("../../../../../config.php");
require_once("../../../../../lib_blocks.php");
require_once("../portfolio2/Route.class.php");
require_once("../role/ratingroles.php");
require_once("../portfolio2/Access/AccessUser.class.php");
require_once("../portfolio2/Check/Check.php");
require_once("../portfolio2/Output/OutputPortfolio.class.php");
require_once("../portfolio2/Output/Tab.class.php");

ini_set( varname: 'display_errors', newvalue: 1);
error_reporting( level: E_ALL);

$cmd = optional_param('cmd', 'controllerportfolio', PARAM_TEXT);
$do = optional_param('do', 'index', PARAM_TEXT);
$id = optional_param('id', '', PARAM_TEXT);

Route::run($cmd, $do, $id);

function debug($var)
{
    echo '<pre>';
    var_dump($var);
    echo '</pre>';
    exit();
}

```

Рисунок 3.7 – структура файла index.php

Здесь происходит подключение всех конфигурационных файлов, файлов классов, файлов библиотек. После того как будут подключены все необходимые файлы для работы модуля, затем будет вызван статический метод run класса Route с необходимыми параметрами:

- \$cmd;
- \$do;
- \$id.

\$cmd – это переменная хранящая название необходимого класса.

\$do – это переменная содержащая название метода класса \$cmd.

\$id – это переменная хранящая уникальный идентификатор.

При первом переходе по ссылке портфолио, создается экземпляр класса Controller.Portfolio.class.php и выполняется метод index. Класс Controller.Portfolio.class.php наследуется от класса CGeneral.class.php который, в свою очередь наследуется от класса GeneralConfig.class.php.

Метод `index` подключает `php` шаблон. Данные фрагменты кода классов представлены в приложении А.

На рисунке 3.8 представлена структура класса `Route`.

```
class Route
{
    public static function Run($cmd, $do, $arg = '')
    {
        if (class_exists($cmd))
        {
            $cmd = new $cmd();
            if (method_exists($cmd, $do))
            {
                $cmd->$do($arg);
                die();
            } else
            {
                echo 'method<b>' . $do . '</b> is not exist';
            }
        } else
        {
            echo 'Class <b>' . $cmd . '</b> is not exist';
        }
    }
}
```

Рисунок 3.8 – структура класса `Route`

Весь код классов паттерна MVC представлен в приложении Б.

Класс `Route` необходим для того, чтобы запускать нужный конструктор в зависимости от того какой пришел запрос от пользователя. Запрос разбивается на составляющие:

- класс;
- метод;
- идентификатор.

В методе `Run` происходит проверка на существование данного класс, если класс существует, то создаем экземпляр данного класса. Затем происходит проверка на существование данного метода, если метод существует, то происходит его выполнение, в противном же случаи будет выдано сообщение об ошибке.

На листинге 3.1 представлен метод конструктора, который отвечает за отрисовку доступных категорий рейтинга эффективности программ.

### Листинг 3.1 – функция «setTabCategory»

```
/**
 * This method set tabs of categories
 */
protected function setTabCategory()
{
    $accessCategoryTab = $this->accessUser->getAccessCategoryTab();
    if (Check::isExistCategory($accessCategoryTab))
    {
        $arrayTabsCategory = $this->modelPortfolio-
>getTabsCategory($accessCategoryTab);
        $this->paramView = array('arrayTabsCategory' => $arrayTabsCategory);
    } else
    {
        $this->render('View/error/errorCategoryNotFound', array('error' => 'Нет
доступных категорий'));
        die();
    }
}
}
```

На листинге 3.2 продемонстрирован фрагмент кода который формирует доступные вкладки портфолио.

### Листинг 3.2 – функция «setTabPortfolio»

```
/**
 * This method set all access tabs for portfolio by $categoryId
 * @param int $categoryId id category
 */
public function setTabPortfolio($categoryId)
{
    $accessPortfolioTab = $this->accessUser-
>getAccessPortfolioTab($categoryId);

    if (Check::isExistAccessPortfolioTab($accessPortfolioTab))
    {
        $arrayTabsPortfolio = $this->modelPortfolio-
>getTabsPortfolio($accessPortfolioTab);
        OutputPortfolio::printTabs($arrayTabsPortfolio, 'TabPortfolio');
    } else
    {
        echo '<b>Нет доступных вкладок портфолио</b><br>';
    }
}
}
```

Так же в портфолио рейтинга эффективности работы есть иерархическое дерево со всеми возможными критериями. Данная рекурсивная функция будет выполняться до тех пор, пока условие выхода из функции не будет соблюдено.

Полный код данного файла tree.php представлен в приложении В.

### 3.2.2 Модуль справочники

Справочников для портфолио рейтинга эффективности работы были созданы на основе плагина jgGrid. Данный плагин работает на под управлением JavaScript библиотеки JQuery. Это довольно мощный плагин для создания разного рода таблиц в веб представлении. Его возможности позволяют создавать не только простые таблицы, но и таблицы с вложенными таблицами, а также создавать деревья.

Плагин jgGrid использует Ajax-вызовы для получения данных от сервера и наоборот. Изменение информации на сервере происходит с помощью PHP, Java или других серверных языков.

Таблица jgGrid представляет из себя четыре компонента:

- заголовок;
- заголовок колонок;
- тело;
- панель навигации.

На рисунке 3.9 представлен пример jgGrid таблицы

id	Наименование
1	Магистратура
2	Аспирантура
3	ППС
4	Кафедры
5	Научный сотрудник
6	Научные подразделения
7	Институты/факультеты
8	Преподаватель СПО

Рисунок 3.9 – Таблица jgGrid

Заголовок – это верхний компонент таблицы, хранящий в себе информацию об отображаемых данных.

Заголовки колонок – содержат в себе информацию о колонках.

Тело – данный компонент получает данные от сервера и отображает их в соответствии с настройками в colModel.

Навигация – этот компонент позволяет перемещаться между страницами таблицами, также навигация содержит кнопки просмотра, редактирования, добавления, удаления. Библиотека jgGrid позволяет создавать собственные кнопки, которые будут выполнять нужные функции.

Настройки colModel отвечают за отображение таблиц. Обязательное поле, которое должно быть присуще каждой jgGrid таблицы – это поле name. Данное поле отвечает за отображение данных приходящих с сервера в таблице.

Некоторые свойства colModel:

- index;
- label;
- resizable;
- search;
- sortable.

Свойство index – позволяет устанавливать имя индекса при сортировке. Передаётся в качестве параметра и представляет из себя String переменную.

Свойство label – если ответ от сервера приходит пустым, данное поле определяет заголовок для этого поля. Если же colNames пуст и свойство label тоже, то берётся заголовок для этой строки из свойства name.

Свойство resizable – отвечает за то, что, можно ли изменять ширину колонки.

Свойство search – запрещает или разрешает поиск по колонке.

Свойство sortable – запрещает или размещает сортировку по колонке.

Данная библиотека позволяет сортировать данные таблицы простым нажатием по заголовку колонке. Программный код справочников представлен в приложении Г.

### 3.3 Тестирование

Подсистема рейтинга эффективности работы расположена в правой части системы «ИнфоБелГУ: Учебный процесс» и представлена на рисунке 3.10.

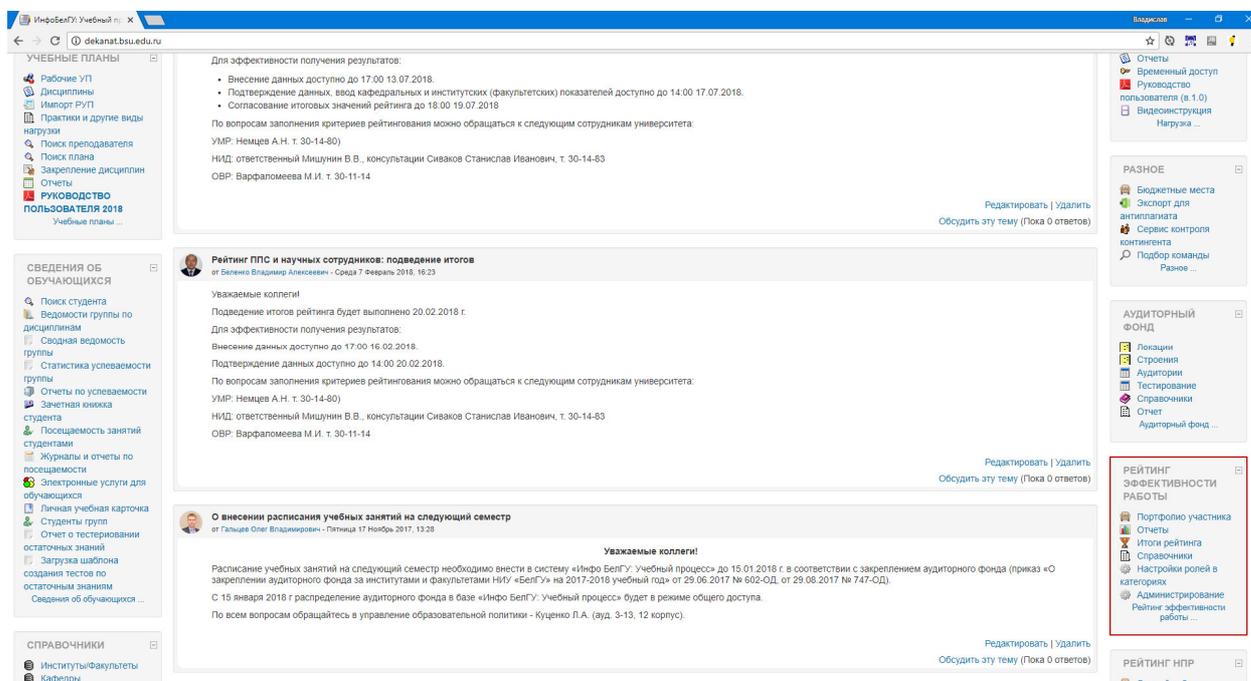


Рисунок 3.10 – Блок рейтинга эффективности работы

Данный блок состоит из нескольких модулей, не все модули доступны всем пользователям системы «ИнфоБелГУ: Учебный процесс». Доступы напрямую связаны с ролями пользователя.

Модули рейтинга эффективности работ:

- портфолио;
- отчеты
- итоги рейтинга;
- справочники;
- настройки ролей в категории.

Модуль «Портфолио» – это своеобразный аналог резюме, оно включает в себя перечень добавленных работ для каждого пользователя.

Модуль «Отчеты» – хранит в себе перечень доступных отчетов, данный перечень будет пополняться.

Модуль «Итоги рейтинга» – отображает итоги за каждый период рейтингования.

Модуль «Справочники» – необходимы для администрирования рейтинга эффективности работы. Они позволяют оперировать категориями, критериями, видами деятельности, периодами и полями критериев рейтинга.

Модуль «настройки ролей в категории» – отвечает за настройки прав доступа для каждой категории.

На рисунке 3.11 представлен модуль «портфолио» рейтинга эффективности работы.

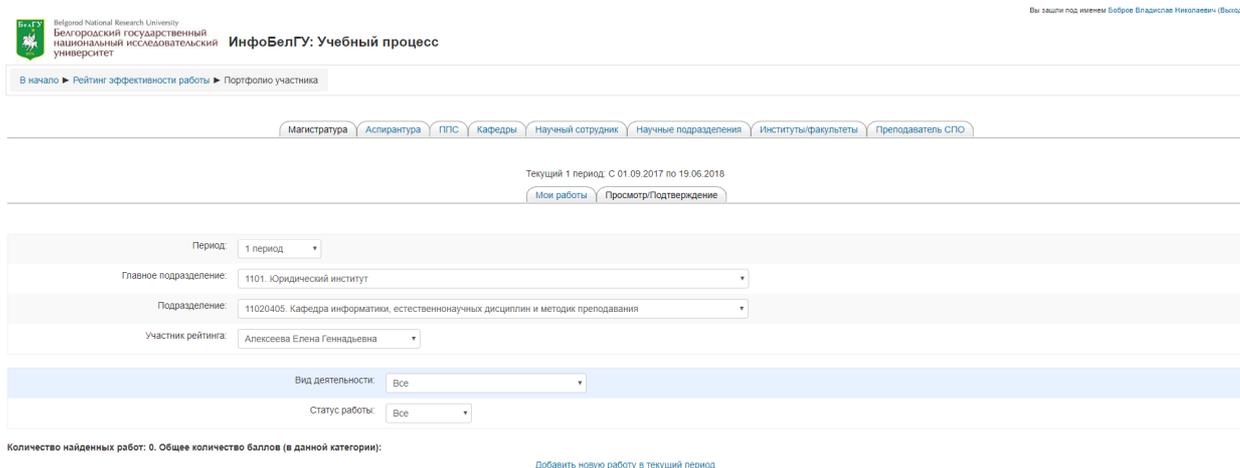


Рисунок 3.11 – Портфолио рейтинга эффективности работы

Как видно на рисунке 3.11 у данного пользователя не добавлено не одной работы. Что бы добавить новую работу нужно нажать на ссылку «Добавить новую работу в текущий период». После чего страница с портфолио закроется и загрузится новая страница, содержащая перечень доступных критериев для категории «Магистратура».

Для каждой категории свои уникальные критерии, которые были утверждены в «положении о рейтинге». Данный документ согласовывается почти со всеми проректорами «НИУ БелГУ».

На рисунке 3.12 представлен перечень доступных критериев для категории «Магистратура».

Категория критериев рейтинга: Магистратура

Участник рейтинга: Алексеева Елена Геннадьевна

Вид деятельности: Выберите наименование вида деятельности ... ▾

Далее >>

[раскрыть все записи](#)

Учебная деятельность (ответственный - Шумакова И.А., тел. 30-10-35)

- 1.1. Выбор тематики исследования (выписка из протокола заседания выпускающей кафедры) [1]
- 1.2. Определение цели, объекта, предмета и задачи исследования (копия индивидуального плана работы магистранта с отметкой руководителя о выполнении данного критерия) [1]
- 1.3. Подготовка и представление теоретико-методологического анализа исследуемой проблемы (рецензия научного руководителя) [5]
- 1.4. Подготовка и представление практической части исследования (рецензия научного руководителя) [7]
- 1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения)
- 1.6. Профессиональная переподготовка, повышение квалификации (сертификат, свидетельство) [6]
- 1.7. Участие в различных профессионально-педагогических олимпиадах, конкурсах (организатор, член жюри) (грамота, диплом) [3]
- 1.8. Призовые места в различных профессионально-педагогических олимпиадах, конкурсах (грамота, диплом) [5]

Научно-исследовательская деятельность (ответственный - Шумакова И.А., тел. 30-10-35)

- 2.1. Публикации научных материалов по теме исследований
- 2.2. Участие в организации тренинга, мастер-класса, круглого стола и т.п. (сертификат) [3]
- 2.3. Участие в организации научной конференции (грамота, диплом) [3]
- 2.4. Участие в работе секций (выступление) на научной конференции, семинаре, форуме, симпозиуме (диплом, сертификат)
- 2.5. Участие в научных проектах (грантах), которые получили финансовую поддержку (справка, подтверждающая участие)
- 2.6. Участие в программах академической мобильности (стажировка) [15]
- 2.7. Получение патента Российской Федерации на изобретение, на полезную модель, на промышленный образец, на селекционное достижение (копия) [10]
- 2.8. Получение свидетельства о государственной регистрации программы для ЭВМ или базы данных (копия) [8]
- 2.9. Победитель стипендиальных программ, научных конкурсов (копия приказа) [5]

Далее >>

Рисунок 3.12 – Иерархический список критериев

Пользователь может выбрать один из доступных критериев рейтингования. Так же есть возможность раскрыть все записи или скрыть все записи, которые имеют уровни вложенности. Пример раскрытия критерия «1.5 Владения иностранным языком» представлен на рисунке 3.13.

- 1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения)
  - 1.5.1. Уровень A1 [2]
  - 1.5.2. Уровень A2 [4]
  - 1.5.3. Уровень B1 [6]
  - 1.5.4. Уровень B2 [8]
  - 1.5.5. Уровень C1 [10]
  - 1.5.6. Уровень C2 [12]

Рисунок 3.13 – Критерий рейтинга

На против каждого критерия в квадратных скобках указан его вес или же количество баллов.

После того как был выбран один из критериев, например 1.5.6 Уровень С2 и нажата кнопка «Далее», откроется новое окно с формой для заполнения полей данного критерия, представленного на рисунке 3.14.

В начало ► Рейтинг эффективности работы ► Портфолио участника ► Добавление новой работы

ID критерия 11 :	1.5.6. Уровень С2
Дата добавления:	14.06.2018
Наименование сертификата	<input type="text"/>
Место выдачи	<input type="text"/>
Дата получения	14.06.2018
Загрузить файл с подтверждающим документом: (размер файла не должен превышать 4Mb)	<input type="button" value="Выберите файл"/> Файл не выбран
Количество авторов:	1 ▾
Автор:	Алексеева Елена Геннадьевна
Вклад автора в работу:	<input type="text" value="100"/> %

Рисунок 3.14 – Форма заполнения критерия

Для того что бы работа была добавлена в портфолио рейтинга эффективности работы необходимо заполнить все поля для данного критерия. Если у работы несколько авторов, то необходимо выбрать нужное количество из выпадающего списка, представленного на рисунке 3.15

Количество авторов:

- 1 ▾
- 1
- 2
- 3
- 4
- 5
- 6
- 7

Автор: \_\_\_\_\_ ва Елен

Вклад автора в работу: \_\_\_\_\_

Рисунок 3.15 – Выпадающий список с количество авторов

После выбора нужного количества авторов, появится новое поле, которое необходимо будет заполнить. Что бы упростить и исключить ошибки при вводе данных было сделано автокомплит для поля «Внутренний соавтор». Все что требуется от пользователя так это начать вводить инициалы соавтора, система сама предложит выбрать нужного или заполнить данное поле собственноручно. На рисунке 3.16 представлено поле «Внутренний соавтор».

Начните вводить ФИО соавтора. Выберите соавтора из предлагаемых вариантов в выпадающем списке.  
Если ФИО будет введено вручную, то соавтор не будет определен системой и дальнейшее заполнение будет невозможно.  
Так же просим обратить внимание, что в качестве соавторов вы можете добавить только участников рейтинга из вашей категории, а баллы за заполнение критерия будут разделены между заполняющим и соавторами в указанном процентном соотношении

Внутренний соавтор: Ad

Вклад автора в работу: %

Абакумова Алина Сергеевна  
Абашкин Игорь Игоревич  
Абашкина Юлия Павловна

ДОБАВИТЬ

Рисунок 3.16 – Поле «внутренний соавтор»

Затем необходимо распределить количество процента вкладов в работу между соавторами, оно не должно превышать 100%. В противном же случаи система не даст добавить такую работу. На рисунке 3.17 представлено распределение процента вклада авторов в работу.

Вклад автора в работу: 90 %

Начните вводить ФИО соавтора. Выберите соавтора из предлагаемых вариантов в выпадающем списке.  
Если ФИО будет введено вручную, то соавтор не будет определен системой и дальнейшее заполнение будет невозможно.  
Так же просим обратить внимание, что в качестве соавторов вы можете добавить только участников рейтинга из вашей категории, а баллы за заполнение к

Внутренний соавтор: Абакумова Алина Сергеевна

Вклад автора в работу: 10 %

ДОБАВИТЬ

Рисунок 3.17 – Распределение процента вклада авторов в работу

После выполненных действий можно будет добавить работу в портфолио рейтинга эффективности работы.

На рисунке 3.18 представлено сообщение об успешном добавлении работы в портфолио.

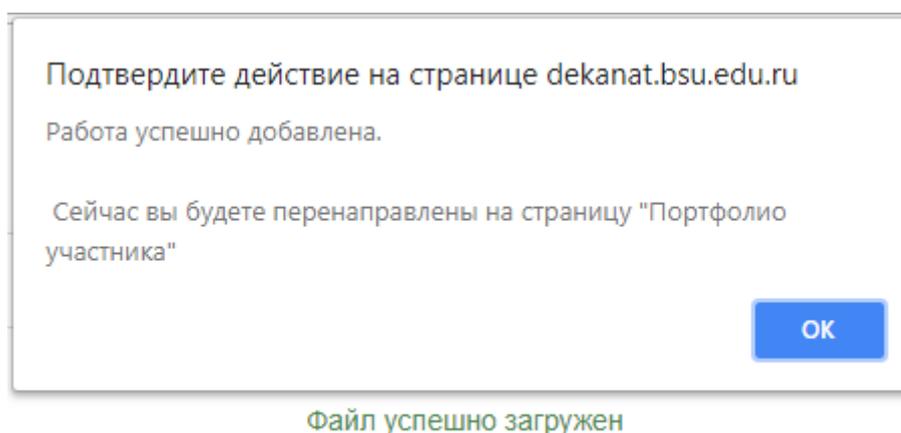


Рисунок 3.18 – Сообщение об успешном добавлении работы в портфолио рейтинга эффективности работы

Теперь после успешного добавления работы. У пользователя в портфолио будет отображена добавленная работа, представленная на рисунке 3.19.

Количество найденных работ: 1. Общее количество баллов (в данной категории): 0.00

5 - 1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения)

11 - 1.5.6. Уровень С2

Скрыть работы

Файл	Авторы	Статус	Комментарии	Дата добавления	Дата отправки	Дата подтверждения/отклонения	Последние действия	Баллы	Действия
chatlog.txt	Алексеева Елена Геннадьевна, Абакумова Алина Сергеевна	Добавлена		14.06.2018	00.00.0000	00.00.0000	Алексеева Елена Геннадьевна 14.06.2018	0.00	<input type="checkbox"/> Редактировать работу <input type="checkbox"/> Удалить работу <input type="checkbox"/> Отправить работу

Раскрыть дополнительную информацию

Добавить новую работу в текущий период

Рисунок 3.19 – Портфолио рейтинга эффективности работы

Добавленные работы в портфолио можно будет:

- редактировать работу;
- удалить работу;
- отправить работу.

Редактирование работы подразумевает под собой изменение ранее внесенных данных. После нажатия на кнопку «редактировать работу» откроется новая страница с полями работы. Здесь можно будет изменить значение полей или же загрузить новый файл с подтверждающим документом. Так же можно будет изменить количество авторов, внесших вклад в работу или же полностью поменять состав соавторов. Можно изменить процентное соотношение вклада автора в работу.

На рисунке 3.20 представлена форма редактирования работы.

В начало ► Рейтинг эффективности работы ► Портфолио участника ► Редактирование работы

ID критерия 11 :	5 - 1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения) 11 - 1.5.6. Уровень C2
Дата добавления:	14.06.2018
Наименование сертификата	<input type="text" value="Тестовая работа"/>
Место выдачи	<input type="text" value="БелГУ"/>
Дата получения	<input type="text" value="14.06.2018"/>
Загрузить файл с подтверждающим документом: (размер файла не должен превышать 4Mb)	<input type="button" value="Выберите файл"/> Файл не выбран <a href="#">Ранее загруженный файл: chatlog.txt</a>
Количество авторов:	<input type="text" value="2"/>
Автор:	<input type="text" value="Алексеева Елена Геннадьевна"/>
Вклад автора в работу:	<input type="text" value="80"/> %

Начните вводить ФИО соавтора. Выберите соавтора из предлагаемых вариантов в выпадающем списке.  
Если ФИО будет введено вручную, то соавтор не будет определен системой и дальнейшее заполнение будет невозможно.  
Так же просим обратить внимание, что в качестве соавторов вы можете добавить только участников рейтинга из вашей категории, а баллы за заполнение критерия будут разделены между заполняющим и соавторами в указанном процентном соотношении.

Автор:	<input type="text" value="Абакумова Алина Сергеевна"/>	Вклад автора в работу:	<input type="text" value="20"/>
		%	

СОХРАНИТЬ

Рисунок 3.20 – Редактирование работы

После того как работа была отредактирована она вновь попадёт в портфолио участника рейтинга эффективности работы.

Функция «Удалить работу» – полностью удаляет работу из портфолио.

Функция «Отправить работу» – поменяет статус работы на «отправлено». Затем данная работа должна быть проверена контроллером. В данном случае в роли контроллера выступает заведующий кафедрой магистратуры.

После того как работа была отправлена на проверку, контроллеру отображаются работы во вкладке «Просмотр/Подтверждение». На рисунке 3.21 представлена вкладка «Просмотр/Подтверждение».

Текущий 1 период, С 01.09.2017 по 19.06.2018  
[Мои работы](#) [Просмотр/Подтверждение](#)

---

Период: 1 период

Главное подразделение: 1101. Юридический институт

Подразделение: 11020405. Кафедра информатики, естественнонаучных дисциплин и методик преподавания

Участник рейтинга: Алексеева Елена Геннадьевна

Вид деятельности: Все

Статус работы: Все

Количество найденных работ: 1. Общее количество баллов (в данной категории): 0.00

**5 - 1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения)**  
**11 - 1.5.6. Уровень С2**

[Скрыть работы](#)

Файл	Авторы	Статус	Комментарии	Дата добавления	Дата отправки	Дата подтверждения/отклонения	Последние действия	Баллы	Действия
chatlog.txt	Алексеева Елена Геннадьевна, Абакумова Алина Сергеевна	Отправлена		14.06.2018	14.06.2018	00.00.0000	Бобров Владислав Николаевич 14.06.2018	0.00	<input checked="" type="checkbox"/> Подтвердить работу <input type="checkbox"/> Отклонить работу

[Раскрыть дополнительную информацию](#) [Добавить новую работу в текущий период](#)

Рисунок 3.21 – вкладка «Просмотр/Подтверждение»

Контроллер может скачать и посмотреть загруженный файл подтверждения, посмотреть дополнительную информацию о работе, а так же подтвердить или отклонить работу.

При отклонении работы, контроллер может написать комментарий к работе, для дальнейшего исправления несоответствий. На рисунке 3.22 представлено диалоговое окно комментария.

**Введите комментарий, причину отклонения:**

\*(можно оставить пустым)

Требуется исправление формы данных

Рисунок 3.22 – Окно комментария

После того как работа была отклонена и комментарий был написан. Автор работы может посмотреть данный комментарий к работе и сделать необходимые правка если это необходимо.

При подтверждении работы контроллером, она остаётся в портфолио и получает статус «Подтвержденная работа». Такая работа идёт в учет результата рейтинга.

На рисунке 3.23 представлено портфолио пользователя с подтвержденной работой

Файл	Авторы	Статус	Комментарии	Дата добавления	Дата отправки	Дата подтверждения/отклонения	Последние действия	Баллы	Действия
chatlog.txt	Алексеева Елена Геннадьевна, Абакумова Алина Сергеевна	Подтверждена	Требуется исправление формы данных	14.06.2018	14.06.2018	14.06.2018	Бобров Владислав Николаевич 14.06.2018	10.80	Отклонить работу

Рисунок 3.23 –Портфолио рейтинга эффективности работы

Подтвержденную работу пользователь может отклонить, и тогда она не уйдет в учет результата рейтинга эффективности работы.

Специально для удобства администрирования были разработаны справочники. Справочники рейтингования представлены на рисунке 3.24.

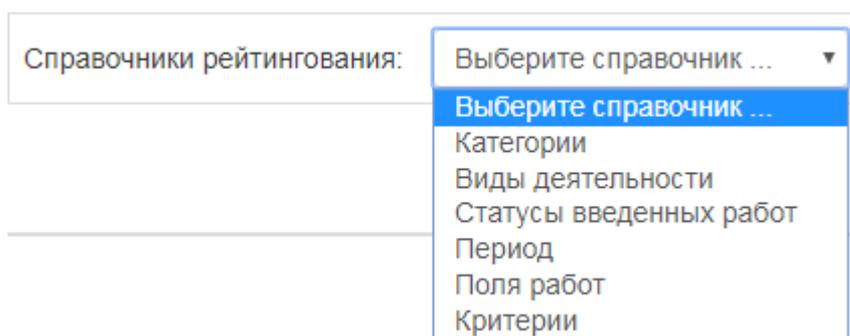


Рисунок 3.24 – Справочники рейтингования

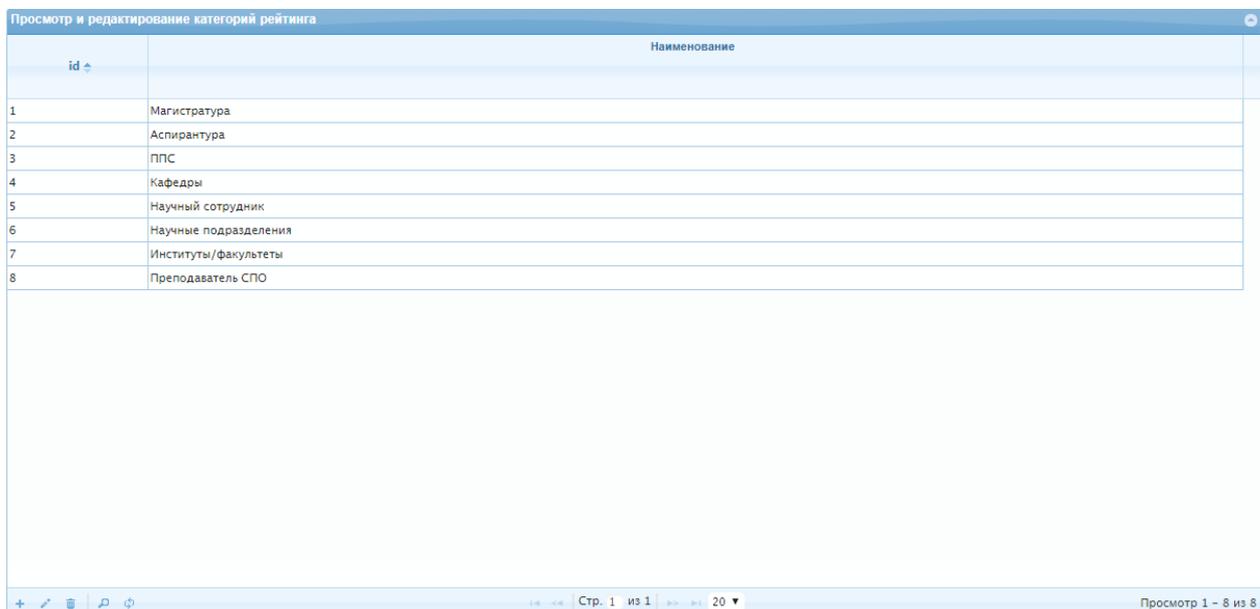
Каждый справочник представляет из себя одну или более таблиц. Каждая таблица формируется динамически на основании данных из базы данных. Так же данные справочники могут не только отображать текущую информацию базы данных, но и добавлять, изменять, удалять, сортировать и осуществлять поиск данных.

Существуют 6 видов справочников:

- категории;
- виды деятельности;
- статусы введенных работ;

- период;
- поля работ;
- критерии.

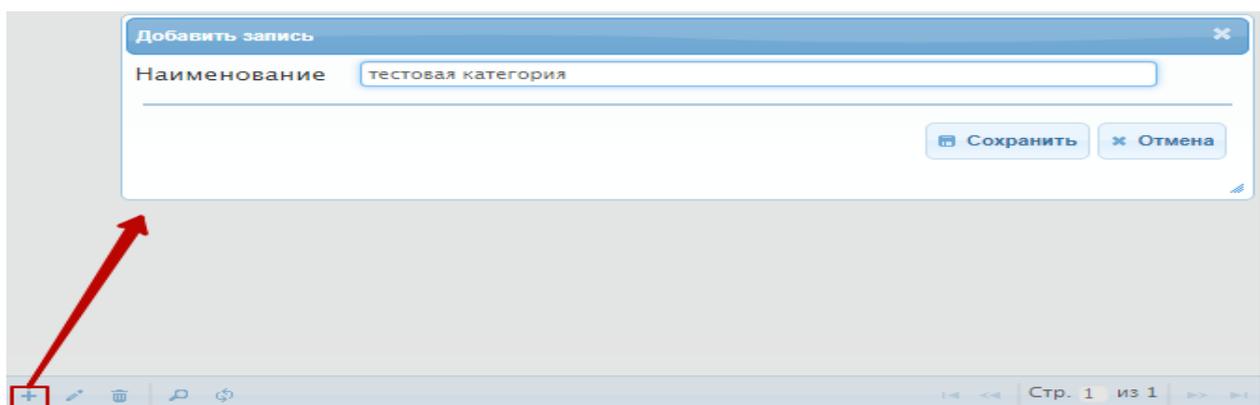
Справочники «Категории» представлены на рисунке 3.25



id ↑	Наименование
1	Магистратура
2	Аспирантура
3	ППС
4	Кафедры
5	Научный сотрудник
6	Научные подразделения
7	Институты/факультеты
8	Преподаватель СПО

Рисунок 3.25 – Справочники «Категории»

В данном справочнике отображаются все доступные категории. Что бы добавить новую категорию, необходимо нажать на кнопку «добавления» представленную на рисунке 3.26.



Добавить запись

Наименование: тестовая категория

Сохранить Отмена

Рисунок 3.26 – Добавление новой категории

После нажатия кнопки сохранить данное наименование новой категории будет добавлено в базу данных. Результат добавления новой категории рейтингования представлен на рисунке 3.26.

Просмотр и редактирование категорий рейтинга	
id	Наименование
1	Магистратура
2	Аспирантура
3	ППС
4	Кафедры
5	Научный сотрудник
6	Научные подразделения
7	Институты/факультеты
8	Преподаватель СПО
9	тестовая категория

Рисунок 3.25 – Результат добавления новой категории

Для наглядного примера работы справочника можно будет изменить вновь добавленную категорию. На рисунке 3.26 представлен процесс редактирования категории. «Тестовая категория» была изменена на «тестовая1».

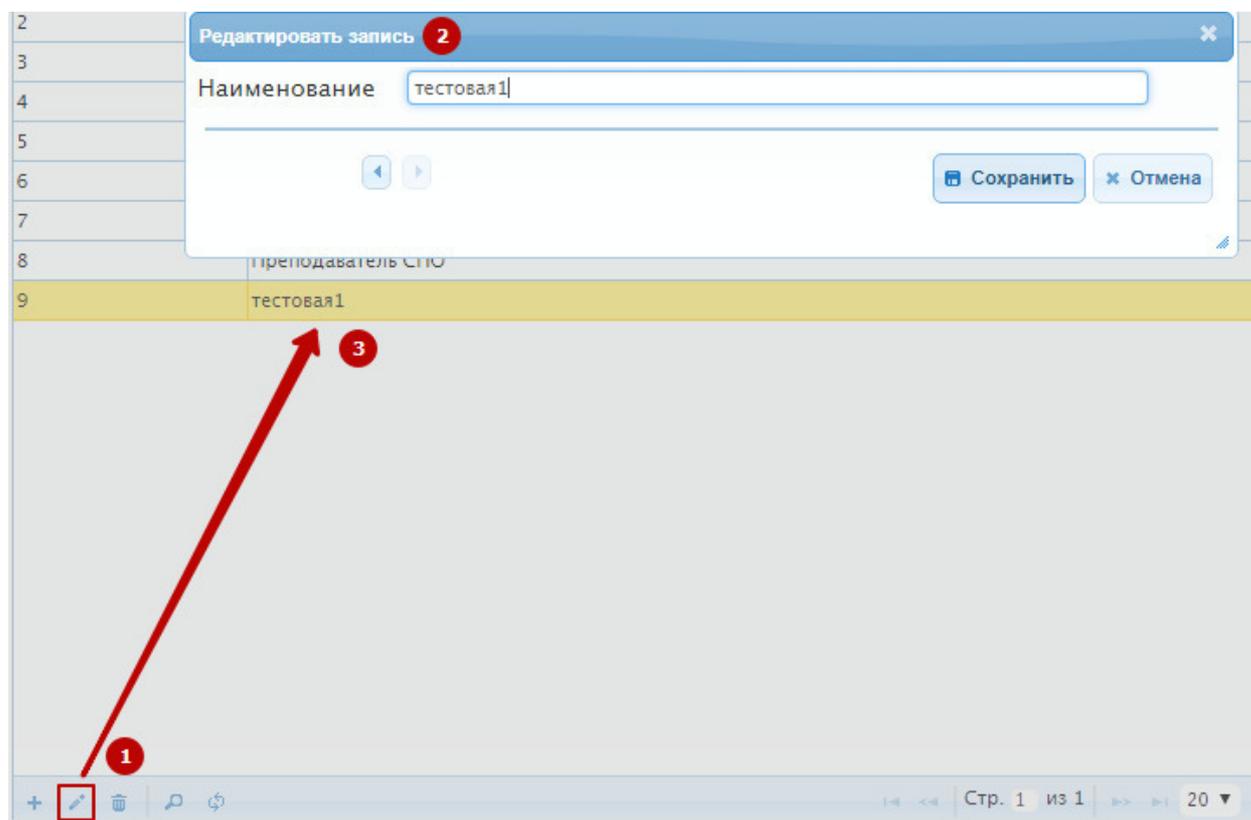


Рисунок 3.26 – Редактирование записи в справочнике «Категории»

Для это потребовалось нажать на кнопку «Редактирования», затем ввести новое значение и нажать копку сохранить.

Что бы удалить выбранную запись необходимо нажать на кнопку удалить и подтвердить данное действие. Процесс удаления представлен на рисунке 3.27.

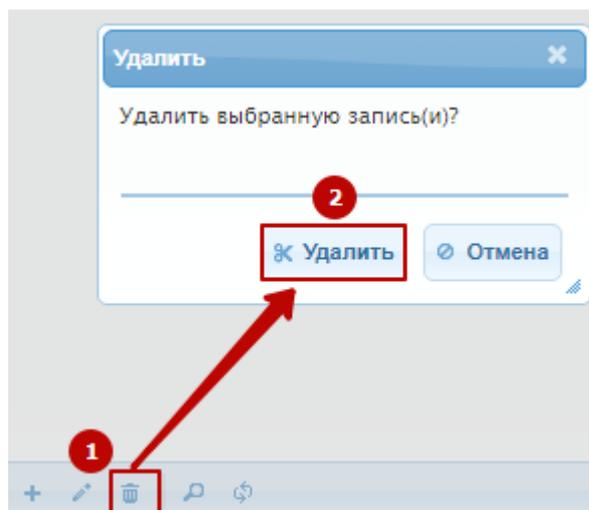


Рисунок 3.27 – Удаление записи из базы данных

Благодаря таким не сложным манипуляциям со справочником, можно легко администрировать системы рейтинга эффективности работы.

Справочники «Виды деятельности», «Статусы введённых работ», «Период» работают аналогичным образом что и справочник «Категории».

На рисунке 3.28 – представлен справочник «Виды деятельности»

Просмотр и редактирование видов деятельности рейтинга	
id ↵	Наименование
1	Учебная деятельность
2	Научно-исследовательская деятельность
3	УМР
4	НИД
5	ОВР

Рисунок 3.28 – Справочник «Виды деятельности»

На рисунке 3.29 представлен справочник «Статусы введенных работ»

Просмотр и редактирование статусов введенных работ рейтинга	
id	Наименование
1	Добавлена
2	Отправлена
3	Подтверждена
4	Отклонена

Стр. 1 из 1      20      Просмотр 1 - 4 из 4

Рисунок 3.29 – Справочник «Статусы введенных работ»

На рисунке 3.30 представлен справочник «Период»

Справочники рейтингования: Период

Учебный год: 17/18

Просмотр и редактирование категорий рейтинга								
id	Учебный год	Название периода	Начало отчетного периода	Окончание отчетного периода	Начало периода подтвержде	Окончание периода подтвер	Кат.ID	Категория
1	18	1 период	2017-09-01	2018-06-19	2017-09-01	2018-06-19	1	Магистратура
11	18	12-й период	2017-09-01	2018-06-19	2017-09-01	2018-06-19	4	Кафедры
12	18	12-й период	2017-09-01	2018-07-19	2017-09-01	2018-07-19	7	Институты/факультеты
14	18	12-й период	2017-08-01	2018-01-19	2017-09-01	2018-01-19	5	Научный сотрудник
45	18	11-й период	2017-02-28	2017-07-20	0000-00-00	0000-00-00	3	ППС
46	18	10 период	2017-09-01	2018-06-19	2017-09-01	2018-06-19	2	Аспирантура
47	18	12-й период	2017-09-01	2018-01-19	2017-09-01	2018-01-19	3	ППС
48	18	12-й период	2017-09-01	2018-01-19	2017-09-01	2018-01-19	6	Научные подразделения
58	18	11-й период	2017-02-28	2017-07-20	0000-00-00	0000-00-00	5	Научный сотрудник
59	18	1 период	2017-07-20	2018-02-27	2017-07-20	2018-02-27	8	Преподаватель СПО
61	18	2 период	2018-03-01	2018-07-29	2018-03-01	2018-07-29	8	Преподаватель СПО
62	18	13-й период	2018-03-11	2018-07-29	2018-03-11	2018-07-29	3	ППС

Стр. 1 из 2      20      Просмотр 1 - 12 из 39

Рисунок 3.30 – Представлен справочник «Период»

Данный справочник показывает все периоды рейтингования за выбранный учебный год.

Справочник «Поля работ» имеет несколько фильтров представленных на рисунке 3.31.

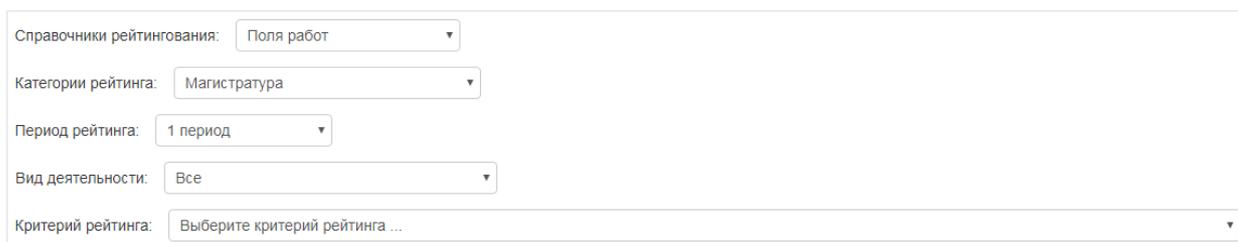


Рисунок 3.31 – Фильтр справочника «Поля работ»

Во-первых, нужно выбрать интересующую категорию.

Во-вторых, необходимо выбрать период данной категории. Так же можно выбрать вид деятельности, но это необязательный фильтр, так как по умолчанию стоит «Все».

В-третьих, самый главный фильтр – это критерий рейтинга.

После того как был выбран критерий рейтинга, загрузиться форма, состоящая из двух вкладок:

- ранее созданные поля;
- новое поле.

Вкладка «Ранее созданные поля» рейтинга эффективности работы состоит из двух таблиц, представленных на рисунке 3.32.



id	Наименование поля на форме ввода	Название поля(отображается для)	Тип поля	Порядок вывода полей на форме
25	Наименование сертификата	name	varchar(512)	1
26	Место выдачи	place	varchar(512)	2
27	Дата получения	update	date	3
2379	Загрузить файл с подтверждающим документом	fileurl	varchar(512)	50

Наименование	Тип поля
<input type="checkbox"/> codesertificate	Числовое целое
<input type="checkbox"/> datesertificate	Дата вещественное
<input type="checkbox"/> Ю курса в СЭО "Тегас"	Числовое целое
<input type="checkbox"/> Ю учебного плана	Одностроннее текстовое
<input type="checkbox"/> ISSN	Одностроннее текстовое
<input type="checkbox"/> ISSN/ISBN	Одностроннее текстовое
<input type="checkbox"/> namesertificate	Одностроннее текстовое
<input type="checkbox"/> Объем средств (руб.)	Денежное значение
<input type="checkbox"/> Библиометрические данные	Одностроннее текстовое
<input type="checkbox"/> ВАК	Флажок выбора
<input type="checkbox"/> Вид документа о квалификации	Список значений
<input type="checkbox"/> Вид занятия, дата и время начала занятия	Многостроннее текстовое
<input type="checkbox"/> Вид издания	Список значений
<input type="checkbox"/> Вид мероприятия	Список значений
<input type="checkbox"/> Вид работы	Список значений
<input type="checkbox"/> Вид участия	Список значений
<input type="checkbox"/> Внеаудит соватории	Одностроннее текстовое
<input type="checkbox"/> Время проведения занятия	Многостроннее текстовое
<input type="checkbox"/> Выходные данные	Многостроннее текстовое
<input type="checkbox"/> Гипотеза	Многостроннее текстовое

Рисунок 3.32 – Вкладка «Ранее созданные поля»

Таблица «Поля работ» отображают поля для ввода данных для выбранного критерия, которые заполняются при добавлении или редактировании работы.

Существует несколько способов добавления новых полей ввода критерия:

- выбор полей из уже существующих полей;
- добавление нового поля.

Выбор полей из уже существующих полей – подразумевает под собой то, что все ранее созданные поля отображаются в таблице «Ранее созданные поля».

На рисунке 3.33 представлена таблица всех существующих полей.

<input type="checkbox"/>	Наименование	Тип поля
<input type="checkbox"/>	codecertificate	Числовое целое
<input type="checkbox"/>	datecertificate	Дата вещественное
<input type="checkbox"/>	ID курса в СЭО "Перас"	Числовое целое
<input type="checkbox"/>	ID учебного плана	Однострочное текстовое
<input type="checkbox"/>	ISSN	Однострочное текстовое
<input type="checkbox"/>	ISSN/ISBN	Однострочное текстовое
<input type="checkbox"/>	namecertificate	Однострочное текстовое
<input type="checkbox"/>	Объём средств (руб.)	Денежное значение
<input type="checkbox"/>	Библиометрические данные	Однострочное текстовое
<input type="checkbox"/>	ВАК	Флажок выбора
<input type="checkbox"/>	Вид документа о квалификации	Список значений
<input type="checkbox"/>	Вид занятия, дата и время начала занятия	Многострочное текстовое
<input type="checkbox"/>	Вид издания	Список значений
<input type="checkbox"/>	Вид мероприятия	Список значений
<input type="checkbox"/>	Вид работы	Список значений
<input type="checkbox"/>	Вид участия	Список значений
<input type="checkbox"/>	Внешние соавторы	Однострочное текстовое
<input type="checkbox"/>	Время проведения занятия	Многострочное текстовое
<input type="checkbox"/>	Выходные данные	Многострочное текстовое
<input type="checkbox"/>	Гипотеза	Многострочное текстовое

Рисунок 3.33 – Таблица «Ранее созданные поля»

На рисунке 3.34 представлен процесс добавления полей.

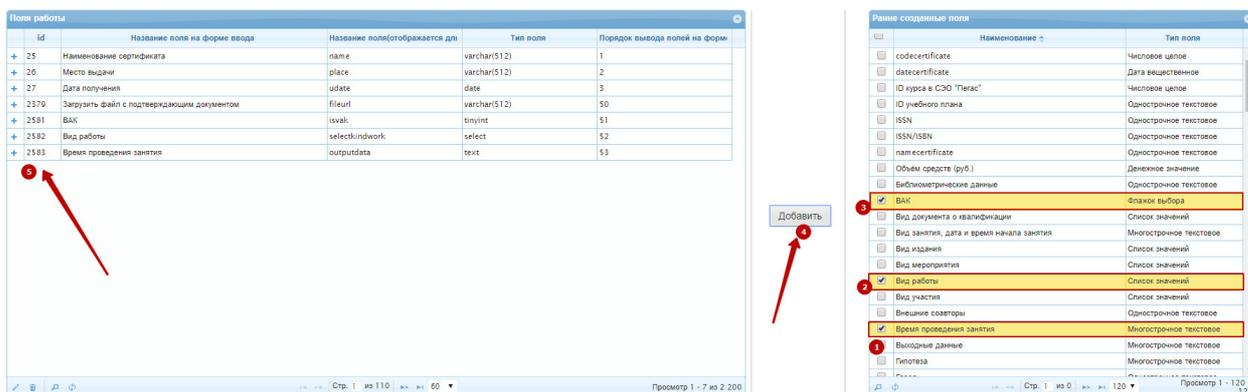


Рисунок 3.34 – Добавление новых полей

Пользователь может выбрать необходимые поля из этой таблицы и затем ему необходимо будет нажать кнопку «Добавить», выбранные поля перенесутся в таблицу «Поля работ» и они же будут отображены на форме добавления или редактирования данных.

Если же нужных полей не оказалось в таблице «Ранее созданные поля», то предусмотрен функционал по добавлению новых полей. На рисунке 3.35 представлена вкладка «Новое поле».

Название поля на форме ввода

Название поля(отображается для разработчиков)

Тип поля

Порядок вывода полей на форме

Рисунок 3.35 – Вкладка «Новое поле»

На данной вкладке расположено несколько полей для ввода данных, а именно:

- название поля на форме ввода;
- название поля (отображаемое для разработчиков);

- тип поля;
- порядок вывода полей на форме.

Поле «Название поля на форме ввода» – отображается пользователю на форме добавления или редактирования работы.

Поле «Название поля» – хранит в себе название переменной. Данное поле напрямую связана с конкретным критерием.

На рисунке 3.36 представлено выпадающий список «Тип поля».

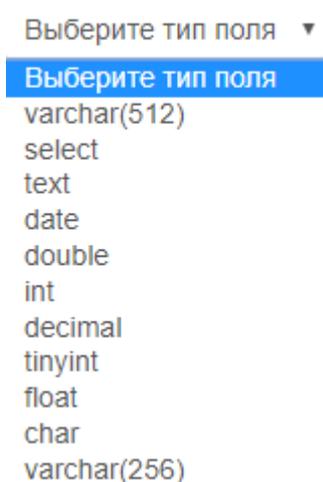


Рисунок 3.36 – Выпадающий список типов полей

Выпадающий список «Тип поля» – хранит в себе все доступные типы полей.

Поле «Порядок вывода полей на форме» – хранит в себе порядковый номер отображения данного поля на форме добавления или редактирования работы.

Справочник рейтинга эффективности работы «Критерии» состоит из фильтров, представленных на рисунке 3.37.

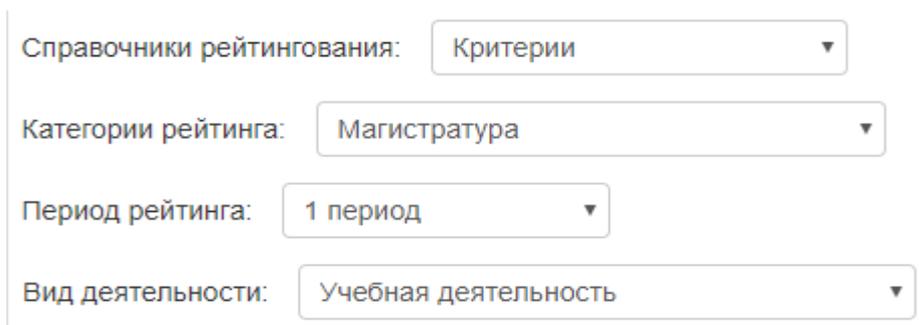


Рисунок 3.37 – Фильтр справочника «Критерии»

После того как были установлены все доступные фильтры, станет доступным таблица справочника «Критерии». На рисунке 3.37 представлена таблица справочника «Критерии».

ID	ID категории	ID вида деятельности	ID родителя	Наименование	type_enter	Путь	onlyone	Дата добавления	Старый ID	Старый ID с	ID периода	Оценка	Нумерация	calcfunc	Формула
1	1	1	0	1.1. Выбор тематики исследования (выписка из протокола заседания выпускающей кафедры)	1	/1		2017-07-25	0	0	1	1	1		
2	1	1	0	1.2. Определение цели, объекта, предмета и задачи исследования (копия индивидуального плана раб	1	/2		2017-07-25	0	0	1	1	2		
3	1	1	0	1.3. Подготовка и представление теоретико-методологического анализа исследуемой проблемы (реци	1	/3		2017-07-25	0	0	1	5	3		
4	1	1	0	1.4. Подготовка и представление практической части исследования (рецензия научного руководителя)	1	/4		2017-07-25	0	0	1	7	4		
5	1	1	0	1.5. Владение иностранным языком (сертификат, подтверждающий уровень владения)	1	/5		2017-07-25	0	0	1	0	5	calcfunc_5	sdid fgbsfgbs
6	1	1	5	1.5.1. Уровень A1	1	/5/6		2017-07-25			1	2	6		
7	1	1	5	1.5.2. Уровень A2	1	/5/7		2017-07-25			1	4	7		
8	1	1	5	1.5.3. Уровень B1	1	/5/8		2017-07-25			1	6	8		
9	1	1	5	1.5.4. Уровень B2	1	/5/9		2017-07-25			1	8	9		
10	1	1	5	1.5.5. Уровень C1	1	/5/10		2017-07-25			1	10	10		
11	1	1	5	1.5.6. Уровень C2	1	/5/11		2017-07-25			1	12	11		
12	1	1	0	1.6. Профессиональная переподготовка, повышение квалификации (сертификат, свидетельство)	1	/12		2017-07-25	0	0	1	6	12		
19	1	1	0	1.7. Участие в различных профессионально-педагогических олимпиадах, конкурсах (организатор, чте	1	/19		2017-07-25	0	0	1	3	13		
28	1	1	0	1.8. Призовые места в различных профессионально-педагогических олимпиадах, конкурсах (грамота,	1	/28		2017-07-25	0	0	1	5	14		

Рисунок 3.37 – Таблица справочника «Критерии»

В данной таблице отображены все доступные критерии для выбранной категории, периода и вида деятельности. Данный справочник позволяет просматривать, добавлять, удалять, редактировать и осуществлять поиск по критериям. Кнопки данных функций находятся в навигационной панели таблицы и отображены на рисунке 3.38.

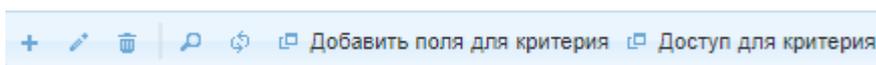


Рисунок 3.38 – Кнопки управления справочником

Так же доступны функции:

- добавление полей для критерия;
- доступ для критерия.

Что бы добавить новое поля для критерия необходимо выделить нужный критерий, а затем нажать на «Добавить поля для критерия». После данного действия откроется новая страница со справочником «Поля работ». Далее необходимо просмотреть записи таблицы «Ранее созданные поля» и выбрать там интересующие поля, если таковых там не оказалось, то нужно перейти на

вкладку «Новое поле». Затем заполнить форму, представленную на рисунке 3.39.

Название поля на форме ввода: Тестовое поле

Название поля(отображается для разработчиков): test

Тип поля: text

Порядок вывода полей на форме: 6

Добавить

Рисунок 3.39 – Форма добавление нового поля для критерия

После заполнения формы необходимо нажать на кнопку «Добавить».

Для того что бы установить доступ критерия необходимо выбрать нужный критерий и нажать на кнопку «Доступ для критерия» в справочной таблицы «Критерии». Откроется диалоговое окно со всеми возможными ролями для данного критерия. На рисунке 3.40 приставленное диалоговое окно с ролями.

ID	Наименование	Запись	Чтение	Удаление	Подтверждение
11	Контролёр УМП (рейтинг обучающихся)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
47	Декан/Директор	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
83	Заведующий кафедрой	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
146	Магистрант	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6062	Начальник отдела аспирантуры	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6189	Начальник отдела аспирантуры	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15764	Сотрудник отдела магистратуры	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Сохранить Отмена

Рисунок 3.40 – Диалоговое окно «Редактировать запись»

Затем нужно выбрать права доступа для ролей участвующих в рейтинге эффективности работы, а именно в категории «Магистратура».

### Выводы по третьему разделу

В данном разделе было произведена программная реализация модуля и его тестирование с подробным описанием всех действий. Был выбран шаблон проектирования, основанный на модели, представлении и контроллере. Так же были выбраны инструментальные средства разработки. Правильный подбор средств разработки позволил упростить сам процесс разработки рейтинга эффективности работы. При тестировании системы рейтингования было подробно описано что получилось в итоге разработки.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была проведена работа, направленная на:

- осуществление проектирования базы данных информационной системы рейтингования преподавателей, аспирантов и магистрантов;
- программную реализацию информационной системы рейтингования преподавателей, аспирантов и магистрантов;
- тестирование информационной системы рейтингования преподавателей, аспирантов и магистрантов.

Основными результатами, полученными при выполнении работы, являются:

- информационная система рейтингования преподавателей, аспирантов и магистрантов, обеспечивающая:
  - ввода данных о результатах деятельности преподавателей, аспирантов и магистрантов;
  - загрузку подтверждающих документов;
  - удаленное подтверждение критериев, соответствующими должностными лицами;
- синхронизацию базы данных информационной системы магистрантов, аспирантов, преподавателей, с информационными системами НИУ «БелГУ»;
- хранение информации о результатах научной и учебно-методической деятельности преподавателей, аспирантов и магистрантов;
- автоматизацию обработки информации о результатах деятельности преподавателей, аспирантов и магистрантов;
- контроль заполнения критериев;
- контроль подтверждения критериев;
- назначение прав на подтверждение критериев;

Таким образом, выпускная квалификационная работа по разработки подсистемы формирования рейтинга эффективности работы преподавателей, аспирантов и магистрантов для системы «ИнфоБелГУ: Учебный процесс» выполнена полностью в соответствии с техническим заданием.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Белгородский государственный национальный исследовательский университет. Официальный сайт. [Электронный ресурс]. Режим доступа: <https://www.bsu.edu.ru/bsu/>
2. Университет сегодня. [Электронный ресурс]. Режим доступа: <https://www.bsu.edu.ru/bsu/info/today/>
3. Васильева Е.Ю., Граничина О.А., Трапицын С.Ю. Рейтинг преподавателей, факультетов и кафедр в вузе: Методическое пособие. [Текст] - СПб.: Изд-во РГПУ им. А.И. Герцена, 2007. - 159 с.
4. Гельфрих А.С. Информационная система рейтингования студентов в вузе. XIX Международная научно-практическая конференция «СОВРЕМЕННЫЕ ТЕХНИКА И ТЕХНОЛОГИИ». Секция 7: Информатика и управление в технических системах [Электронный ресурс]. Режим доступа: <http://www.lib.tpu.ru/fulltext/c/2013/C01/V2/118.pdf>
5. РИТМ.Рейтинг [Электронный ресурс]. Режим доступа: <http://ritm.ispu.ru/>
6. Липатов В.А. Разработка индивидуального рейтинга преподавателей в системе управления качеством образовательного процесса в медицинском вузе: дис. – М.[Высшая школа экономики]
7. «ИнфоБелГУ: Учебный процесс» [Электронный ресурс]. Режим доступа: <http://dekanat.bsu.edu.ru/>
8. Абдикеев, Н.М. Реинжиниринг бизнес-процессов [Текст] / Н.М. Абдикеев, Т.П. Данько, С.В. Ильдеменов, А.Д. Киселев. – Москва: Эксмо, 2013. – 592 с.
9. Белоногов, Г.Г. Языковые средства автоматизированных информационных систем [Текст] / Г.Г. Белоногов, Б.А. Кузнецов. – Москва: Наука, 2012 – 154 с.

10. Пфаффенбергер, Б. HTML, XHTML и CSS. Библия пользователя [Текст] / Б. Пфаффенбергер, С. Шафер, Ч. Уайт, Б. Кароу. – Москва: Диалектика, 2014 – 752 с.
11. Вейтман, В. Программирование для Web / В. Вейтман. – Москва: Диалектика, 2015 – 368 с.
12. Гвоздева, В.А. Основы построения автоматизированных информационных систем [Текст]: учебник / В.А. Гвоздева, И.Ю. Лаврентьева. – Москва: ФОРУМ, 2015 – 320 с.
13. Паблишер, Г. Руководство по кредитному скорингу [Текст] / Г. Паблишер. – Москва: Эксмо, 2015. – 464 с.
14. Коггзолл, Д. PHP 5. Полное руководство [Текст] / Д. Коггзолл. – Москва: Диалектика, 2014. – 752 с.
15. Дубейковский, В.И. Эффективное моделирование с СА ERwin Process Modeler и AllFusion Process Modeler [Текст] / В.И. Дубейковский. – Москва: Диалог-МИФИ, 2013. – 384 с.
16. Ермаков, С.Л. Работа коммерческого банка по кредитованию заемщиков [Текст] / С.Л. Ермаков. – Москва: Паблишер, 2014. – 275 с.
17. Колисниченко, Д.Н. PHP 5 и MySQL 6. Разработка Web-приложений [Текст] / Д.Н. Колисниченко. – СПб: БХВ-Петербург, 2014. – 528 с.
18. Кроливецкая, Л.П., Банковское дело. Кредитная деятельность коммерческих банков [Текст] / Л.П. Кроливецкая, Е.В. Тихомирова. – Москва: изд: КноРус, 2015. – 280 с.
19. Аткинсон, Л. PHP 5. Библиотека профессионала [Текст] / Л. Аткинсон, З. Сураски. – Москва: Вильямс, 2015 – 944 с.
20. Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томсон. – Москва: Вильямс, 2013. – 848 с.
21. Маклаков, С.В. BPwin и ERwin. CASE – средства разработки информационных систем [Текст] / С.В. Маклаков. – Москва: Диалог-МИФИ, 2014. – 256 с.

## ПРИЛОЖЕНИЕ А

### Скрипты для создания базы данных рейтинга

```
DROP TABLE IF EXISTS `mdl_bsu_rating_access_criteria`;
CREATE TABLE `mdl_bsu_rating_access_criteria` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `criteriaid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_rating_criteria по полю id',
  `rolecategoryid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_role по полю id',
  `iswrite` tinyint(4) DEFAULT '0' COMMENT 'разрешение на запись работы, если 0 значить запись запрещена, если 1 то наоборот',
  `isread` tinyint(4) DEFAULT '0' COMMENT 'разрешение на чтение работы, если 0 значить чтение запрещено, если 1 то наоборот',
  `isdelete` tinyint(4) DEFAULT '0' COMMENT 'разрешение на удаление работы, если 0 значить удаление запрещено, если 1 то наоборот',
  `confirm` tinyint(4) DEFAULT '0' COMMENT 'разрешение на подтверждение работы, если 0 значить подтверждение запрещено, если 1 то наоборот',
  PRIMARY KEY (`id`),
  KEY `criteriaidx` (`criteriaid`, `rolecategoryid`),
  KEY `rolecategoryidx` (`rolecategoryid`),
  CONSTRAINT `access_criteria` FOREIGN KEY (`rolecategoryid`) REFERENCES `mdl_bsu_rating_role_category` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `access_criteria_in_criteria_fk` FOREIGN KEY (`criteriaid`) REFERENCES `mdl_bsu_rating_criteria` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=15821 DEFAULT CHARSET=utf8 COMMENT='Права доступа к критериям';

DROP TABLE IF EXISTS `mdl_bsu_rating_archive_role_assignment`;
CREATE TABLE `mdl_bsu_rating_archive_role_assignment` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `periodid` int(10) unsigned NOT NULL DEFAULT '0',
  `roleid` int(10) unsigned NOT NULL DEFAULT '0',
  `contextid` int(10) unsigned NOT NULL DEFAULT '0',
  `userid` int(10) unsigned NOT NULL DEFAULT '0',
  `timemodified` bigint(10) NOT NULL DEFAULT '0' COMMENT 'дата изменения',
  `modifierid` int(10) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `periodx` (`periodid`),
  CONSTRAINT `archive_role_assignment_in_period` FOREIGN KEY (`periodid`) REFERENCES `mdl_bsu_rating_period` (`id`) ON UPDATE CASCADE
```

```
) ENGINE=InnoDB AUTO_INCREMENT=21121 DEFAULT CHARSET=utf8 COMMENT='Архив назначений ролей';
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_category`;
```

```
CREATE TABLE `mdl_bsu_rating_category` (
```

```
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
`name` varchar(255) NOT NULL DEFAULT '' COMMENT 'Название категория рейтинга',
```

```
`idname` varchar(45) DEFAULT NULL COMMENT 'userid, npid, subdepartmentid, departmentcode',
```

```
`mainfilter` varchar(45) DEFAULT NULL,
```

```
`subfilter` varchar(45) DEFAULT NULL,
```

```
`userfilter` varchar(45) DEFAULT NULL,
```

```
`tablename` varchar(45) DEFAULT NULL,
```

```
`contact` varchar(500) DEFAULT '',
```

```
PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8 COMMENT='Категории рейтингования';
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_coworker_np`;
```

```
CREATE TABLE `mdl_bsu_rating_coworker_np` (
```

```
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
`yearid` int(10) DEFAULT '0',
```

```
`userid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_user по полю id',
```

```
`npid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_scientificunit по полю id',
```

```
`positionid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_staffpositions по полю id',
```

```
`stavka` float NOT NULL DEFAULT '0' COMMENT 'зарплатная ставка научного сотрудника',
```

```
`manager` tinyint(4) DEFAULT '0' COMMENT 'исполняющий обязанности...; например: Зав. кафедрой',
```

```
`id_fl` char(36) NOT NULL DEFAULT '' COMMENT 'связь с таблицей mdl_bsu_vw_tb_1csotr по полю id_fl; физическое лицо',
```

```
`id_sotr` char(36) NOT NULL DEFAULT '' COMMENT 'связь с таблицей mdl_bsu_vw_tb_1csotr по полю id_sotr; должность',
```

```
PRIMARY KEY (`id`),
```

```
KEY `yearid` (`yearid`),
```

```
KEY `userid` (`userid`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=85 DEFAULT CHARSET=utf8 COMMENT='Сотрудники научного подразделения';
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_criteria`;
```

```
CREATE TABLE `mdl_bsu_rating_criteria` (
```

```
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```

`categoryid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_rating_category по полю id',

`kindworkid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_rating_kindworking по полю id',

`parentid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с вышестоящим критерием',

`name` text NOT NULL COMMENT 'наименование критерия',

`type_enter` tinyint(4) DEFAULT '0' COMMENT 'тип критерия - ручной ввод, автоматический, полуавтомат',

`path` varchar(255) DEFAULT '/1',

`datein` date DEFAULT NULL,

`oldid` int(10) DEFAULT NULL,

`oldparentid` int(10) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `categoryidx` (`categoryid`),

KEY `parentidx` (`parentid`),

KEY `criteria__in__kindworking__fk` (`kindworkid`),

CONSTRAINT `criteria__in__category__fk` FOREIGN KEY (`categoryid`) REFERENCES
`mdl_bsu_rating_category` (`id`) ON UPDATE CASCADE,

CONSTRAINT `criteria__in__kindworking__fk` FOREIGN KEY (`kindworkid`) REFERENCES
`mdl_bsu_rating_kindworking` (`id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=1082 DEFAULT CHARSET=utf8 COMMENT='Критерии';

DROP TABLE IF EXISTS `mdl_bsu_rating_enter_work`;

/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `mdl_bsu_rating_enter_work` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`criteriaid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_edyear по
полю id',
`statusid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_rating_status_inputdata по полю id',
`fileurl` varchar(255) DEFAULT '' COMMENT 'ссылка на файл с работой',
`comment` text COMMENT 'комментарий к работе',
`datecreate` date NOT NULL DEFAULT '0000-00-00' COMMENT 'дата создания работы',
`datesend` date NOT NULL DEFAULT '0000-00-00' COMMENT 'дата отправления работы',
`dateconfirm` date NOT NULL DEFAULT '0000-00-00' COMMENT 'дата подтверждения работы',
`usermodified` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'пользователь который, изменил
работу',
`timemodified` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'время изменения работы',
`oldid` int(11) DEFAULT '0',
`name` text,

```

`amount` varchar(256) DEFAULT NULL,  
`articlename` varchar(512) DEFAULT NULL,  
`city` varchar(512) DEFAULT NULL,  
`description` text,  
`editionname` varchar(512) DEFAULT NULL,  
`endinternship` date DEFAULT NULL,  
`fullname` varchar(512) DEFAULT NULL,  
`goal` text,  
`hour` double DEFAULT NULL,  
`internship` varchar(512) DEFAULT NULL,  
`issn` varchar(512) DEFAULT NULL,  
`language` varchar(512) DEFAULT NULL,  
`number` varchar(512) DEFAULT NULL,  
`numbersheet` int(11) DEFAULT NULL,  
`object` text,  
`organizer` varchar(512) DEFAULT NULL,  
`place` varchar(512) DEFAULT NULL,  
`prizeplace` varchar(512) DEFAULT NULL,  
`reportname` varchar(512) DEFAULT NULL,  
`startinternship` date DEFAULT NULL,  
`status` varchar(512) DEFAULT NULL,  
`subject` text,  
`timefinancial` varchar(512) DEFAULT NULL,  
`update` date DEFAULT NULL,  
`url` varchar(512) DEFAULT NULL,  
`formwork` varchar(512) DEFAULT NULL,  
`outputdata` text,  
`circulation` int(11) DEFAULT NULL,  
`countexternalcollaborator` int(11) DEFAULT NULL,  
`countissues` int(11) DEFAULT NULL,  
`countprintpage` int(11) DEFAULT NULL,  
`externalcollaborator` text,  
`impaktfactorrinz` float DEFAULT NULL,  
`impaktfactorscopus` float DEFAULT NULL,  
`impaktfactorwebofscience` float DEFAULT NULL,  
`namepublishing` text,  
`namereport` varchar(512) DEFAULT NULL,

`numberdocument` varchar(512) DEFAULT NULL,  
`publishing` varchar(512) DEFAULT NULL,  
`selectcountry` int(11) DEFAULT NULL,  
`selectdegree` int(11) DEFAULT NULL,  
`selectfieldscience` int(11) DEFAULT NULL,  
`selectkindwork` int(11) DEFAULT NULL,  
`selectlanguage` int(11) DEFAULT NULL,  
`selectlevel` int(11) DEFAULT NULL,  
`selectpnr` int(11) DEFAULT NULL,  
`selectscience` int(11) DEFAULT NULL,  
`selectsciencebelgu` int(11) DEFAULT NULL,  
`selectspecial` int(11) DEFAULT NULL,  
`selecttypeofevent` int(11) DEFAULT NULL,  
`selecttypeofparticipation` int(11) DEFAULT NULL,  
`selecttypeprog` int(11) DEFAULT NULL,  
`selecttypepublication` int(11) DEFAULT NULL,  
`selectreward` int(11) DEFAULT NULL,  
`isvak` tinyint(4) DEFAULT NULL,  
`bibliometricdata` char(1) DEFAULT NULL,  
`countcoauthorsmaster` int(11) DEFAULT NULL,  
`countcoauthorspostgraduate` int(11) DEFAULT NULL,  
`countcoauthorsstudent` int(11) DEFAULT NULL,  
`countryexternalcollaborator` char(1) DEFAULT NULL,  
`externalcollaboratorstudent` varchar(512) DEFAULT NULL,  
`financing` varchar(512) DEFAULT NULL,  
`nameproject` varchar(512) DEFAULT NULL,  
`num` varchar(512) DEFAULT NULL,  
`hypothesis` text,  
`codecertificate` int(11) DEFAULT NULL,  
`discipline` varchar(512) DEFAULT NULL,  
`selectevent` int(11) DEFAULT NULL,  
`town` varchar(512) DEFAULT NULL,  
`selectdoctypekvalif` int(11) DEFAULT NULL,  
`ou` text,  
`subdepid` int(11) DEFAULT NULL,  
`planid` varchar(512) DEFAULT NULL,  
`umkdid` int(11) DEFAULT NULL,

`disciplineid` int(11) DEFAULT NULL,  
`disserid` int(11) DEFAULT NULL,  
`dissername` text,  
`disserauthor` varchar(512) DEFAULT NULL,  
`testtest` varchar(512) DEFAULT NULL,  
`numberpoint` float DEFAULT NULL,  
`numberstudent` int(11) DEFAULT NULL,  
`numberprogram` int(11) DEFAULT NULL,  
`numberforeigncitizensnpr` float DEFAULT NULL,  
`totalnumbernpr` float DEFAULT NULL,  
`totalnumberfaculty` float DEFAULT NULL,  
`numberforeignteachers` int(11) DEFAULT NULL,  
`numberdisciplinewithoutumk` int(11) DEFAULT NULL,  
`numberchairs` int(11) DEFAULT NULL,  
`numbernprwithdegree` float DEFAULT NULL,  
`percentyoungnpr` float DEFAULT NULL,  
`numberworkers` int(11) DEFAULT NULL,  
`numberaspirants` int(11) DEFAULT NULL,  
`totalnumberapprovedscientificareas` int(11) DEFAULT NULL,  
`totalnumberscientificschools` int(11) DEFAULT NULL,  
`amountmagistaspirant` float DEFAULT NULL,  
`totalnumberstudent` int(11) DEFAULT NULL,  
`numberstudentwithdiplom` int(11) DEFAULT NULL,  
`numberstudentfirstgraduate` int(11) DEFAULT NULL,  
`numberforeignstudents` int(11) DEFAULT NULL,  
`numberstudentsfromsng` int(11) DEFAULT NULL,  
`numberstrudentsaducationabroad` int(11) DEFAULT NULL,  
`totalnumberfulltimestudent` int(11) DEFAULT NULL,  
`numberfulltimeforeignstudents` int(11) DEFAULT NULL,  
`numberforeigncitizenwithoutaspirant` int(11) DEFAULT NULL,  
`totalnumberaspirant` int(11) DEFAULT NULL,  
`numberaspirantsng` int(11) DEFAULT NULL,  
`amountmoney` decimal(10,0) DEFAULT NULL,  
`numberemployedgraduates` int(11) DEFAULT NULL,  
`totalnumbergraduates` int(11) DEFAULT NULL,  
`numberevent` int(11) DEFAULT NULL,  
`numberinstitute` int(11) DEFAULT NULL,

```

`numberscientificunit` int(11) DEFAULT NULL,
`numberscientificprojects` int(11) DEFAULT NULL,
`amountmoneyRID` decimal(10,0) DEFAULT NULL,
`numberproject` int(11) DEFAULT NULL,
`placeinchallenge` int(11) DEFAULT NULL,
`numberviolations` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `criteriaidx` (`criteriaid`),
KEY `oldidx` (`oldid`),
KEY `enter_work__in__status_inputdata__fk` (`statusid`),
CONSTRAINT `enter_work__in__criteria__fk` FOREIGN KEY (`criteriaid`) REFERENCES
`mdl_bsu_rating_criteria` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `enter_work__in__status_inputdata__fk` FOREIGN KEY (`statusid`) REFERENCES
`mdl_bsu_rating_status_inputdata` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=119196 DEFAULT CHARSET=utf8 COMMENT='Добавление
работы';

DROP TABLE IF EXISTS `mdl_bsu_rating_enter_work_mark`;
CREATE TABLE `mdl_bsu_rating_enter_work_mark` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`enterworkid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_rating_enter_work по полю id',
`userid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_user по полю id',
`departmentcode` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_ref_department по полю Id',
`subdepartmentid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_vw_ref_subdepartments по полю id',
`npid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_scientificunit по
полю id',
`groupid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_groups по
полю id',
`percent` float NOT NULL DEFAULT '0' COMMENT 'процент оценки работы',
`mark` float NOT NULL DEFAULT '0' COMMENT 'оценка работы',
`oldid` int(11) DEFAULT '0',
PRIMARY KEY (`id`),
KEY `enterworkidx` (`enterworkid`),
KEY `oldidx` (`oldid`),
KEY `useridx` (`userid`),
CONSTRAINT `enter_work_mark__in__enter_work__fk` FOREIGN KEY (`enterworkid`) REFERENCES
`mdl_bsu_rating_enter_work` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=136047 DEFAULT CHARSET=utf8 COMMENT='Оценка работы';

```

```

DROP TABLE IF EXISTS `mdl_bsu_rating_fielddata`;
CREATE TABLE `mdl_bsu_rating_fielddata` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`criteriaid` int(10) unsigned NOT NULL COMMENT 'связь с таблицей mdl_bsu_rating_criteria по полю id',
`name` varchar(255) NOT NULL DEFAULT '' COMMENT 'название поля на форме ввода(отображается для пользователя)',
`fieldname` varchar(190) NOT NULL DEFAULT '' COMMENT 'название поля в таб. mdl_bsu_rating_entered_data (отображается для разработчиков)',
`fieldtype` varchar(255) NOT NULL DEFAULT '' COMMENT 'тип для поля fieldname',
`ordernum` smallint(6) NOT NULL DEFAULT '0' COMMENT 'порядок вывода полей на форме',
`descriptionfield` text COMMENT 'описание для полей критериев',
PRIMARY KEY (`id`),
KEY `criteriaidx` (`criteriaid`),
KEY `fieldname` (`fieldname`),
CONSTRAINT `fielddata__in__criteria_fk` FOREIGN KEY (`criteriaid`) REFERENCES `mdl_bsu_rating_criteria` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=2578 DEFAULT CHARSET=utf8 COMMENT='Поля данных по работам';

```

```

DROP TABLE IF EXISTS `mdl_bsu_rating_kindworking`;
CREATE TABLE `mdl_bsu_rating_kindworking` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(255) NOT NULL DEFAULT '' COMMENT 'наименование группы рейтинга',
`contact` varchar(500) DEFAULT '',
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8 COMMENT='Виды деятельности';

```

```

DROP TABLE IF EXISTS `mdl_bsu_rating_mark_criteria`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `mdl_bsu_rating_mark_criteria` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`periodid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_rating_period по полю id',
`criteriaid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_rating_criteria по полю id',
`mark` float NOT NULL COMMENT 'оценка в виде баллов',
`sortnumber` smallint(6) DEFAULT '0',

```

```

`calcfunc` varchar(45) DEFAULT NULL,
`formula` varchar(255) DEFAULT NULL,
`onlyone` int(10) DEFAULT NULL COMMENT 'Уникальный критерий: 1 - критерий не заполнен; 0 -
критерий заполнен. null - обычный критерий',
PRIMARY KEY (`id`),
KEY `periodidx` (`periodid`),
KEY `mdl_bsu_rating_mark_criteria_criteriaaid_index` (`criteriaid`),
KEY `criteriaidx` (`criteriaid`),
CONSTRAINT `mark_criteria_in_criteria_fk` FOREIGN KEY (`criteriaid`) REFERENCES
`mdl_bsu_rating_criteria` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `mark_criteria_in_period_fk` FOREIGN KEY (`periodid`) REFERENCES
`mdl_bsu_rating_period` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1379 DEFAULT CHARSET=utf8 COMMENT='Оценки по
критериям';

DROP TABLE IF EXISTS `mdl_bsu_rating_period`;
CREATE TABLE `mdl_bsu_rating_period` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`yearid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_edyear по
полю Id',
`nameperiod` varchar(255) NOT NULL DEFAULT '' COMMENT 'наименование периода',
`startperiod` date NOT NULL DEFAULT '0000-00-00' COMMENT 'дата начала периода рейтингования',
`endperiod` date NOT NULL DEFAULT '0000-00-00' COMMENT 'дата конца периода рейтингования',
`startperiodconfirm` date DEFAULT '0000-00-00' COMMENT 'дата начала подтверждения периода',
`endperiodconfirm` date DEFAULT '0000-00-00' COMMENT 'дата конца подтверждения периода',
`categoryid` int(10) unsigned DEFAULT NULL,
`time` datetime DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `yearidx` (`yearid`),
KEY `categoryidx` (`categoryid`),
CONSTRAINT `categoryfk` FOREIGN KEY (`categoryid`) REFERENCES `mdl_bsu_rating_category` (`id`) ON
DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=63 DEFAULT CHARSET=utf8 COMMENT='Периоды
рейтингования';

DROP TABLE IF EXISTS `mdl_bsu_rating_referencedata_field`;
CREATE TABLE `mdl_bsu_rating_referencedata_field` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`fieldname` varchar(190) NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_rating_fielddata по
полю id',
`name` varchar(255) NOT NULL DEFAULT '' COMMENT 'наименование справочных данных',

```

```

`notusing` tinyint(4) DEFAULT '0',
`oldid` int(10) DEFAULT '0',
`oldtablename` varchar(100) DEFAULT "",
`oldfieldname` varchar(100) DEFAULT "",
PRIMARY KEY (`id`),
KEY `fieldname` (`fieldname`)
) ENGINE=InnoDB AUTO_INCREMENT=708 DEFAULT CHARSET=utf8 COMMENT='Справочные данные
к полям';

```

```

DROP TABLE IF EXISTS `mdl_bsu_rating_result`;
CREATE TABLE `mdl_bsu_rating_result` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`yearid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_edyear по
полю Id',
`periodid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_rating_period по
полю id',
`userid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_user по полю id',
`departmentcode` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_ref_department по полю Id',
`subdepartmentid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_vw_ref_subdepartments по полю id',
`npid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_scientificunit по
полю id',
`groupid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_bsu_ref_groups по
полю id',
`totalmark` float NOT NULL DEFAULT '0' COMMENT 'итоговая оценка',
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='Итоги рейтинга';

```

```

DROP TABLE IF EXISTS `mdl_bsu_rating_role_category`;
CREATE TABLE `mdl_bsu_rating_role_category` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`categoryid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей
mdl_bsu_rating_category по полю id',
`roleid` int(10) unsigned NOT NULL DEFAULT '0' COMMENT 'связь с таблицей mdl_role по полю id',
`iscontrole` tinyint(4) DEFAULT '0' COMMENT '1 - признак обозначающий что, есть ли у данной
роли(roleid) из таблицы(mdl_role) подчиненные роли. В противном же случаи 0',
`subroleids` varchar(45) DEFAULT "" COMMENT 'подчиненные роли для roleid',
PRIMARY KEY (`id`),
KEY `categoryidx` (`categoryid`,`roleid`),
KEY `roleidx` (`roleid`),

```

```
CONSTRAINT `role_category_in_category_fk` FOREIGN KEY (`categoryid`) REFERENCES
`mdl_bsu_rating_category` (`id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=204 DEFAULT CHARSET=utf8 COMMENT='Какие роли к каким
категориям относятся';
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_status_inputdata`;
CREATE TABLE `mdl_bsu_rating_status_inputdata` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(255) NOT NULL DEFAULT '' COMMENT 'наименование статуса',
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8 COMMENT='Статусы введенных
данных';
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_umkd`;
CREATE TABLE `mdl_bsu_rating_umkd` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`category_id` int(10) unsigned NOT NULL,
`pegas_course_id` int(11) NOT NULL DEFAULT '0',
`name` text COLLATE utf8_unicode_ci,
`pages` double NOT NULL DEFAULT '0',
`hours` double NOT NULL DEFAULT '0',
`film_type` text COLLATE utf8_unicode_ci,
`film_hours` double NOT NULL DEFAULT '0',
`date_input` date DEFAULT NULL,
`work_id` int(10) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `pegasid` (`pegas_course_id`)
) ENGINE=InnoDB AUTO_INCREMENT=77 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_umkd_authors`;
CREATE TABLE `mdl_bsu_rating_umkd_authors` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`umkdid` int(11) NOT NULL DEFAULT '0',
`fio` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
`email` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
`part` double NOT NULL DEFAULT '0',
`linked` int(11) DEFAULT NULL,
`userid` bigint(10) DEFAULT NULL,
PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=96 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
DROP TABLE IF EXISTS `mdl_bsu_rating_umkd_authors_films`;
```

```
CREATE TABLE `mdl_bsu_rating_umkd_authors_films` (
```

```
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
`umkdid` int(11) NOT NULL DEFAULT '0',
```

```
`fio` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
```

```
`email` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
```

```
`part` double NOT NULL DEFAULT '0',
```

```
`linked` int(11) DEFAULT NULL,
```

```
`userid` int(11) DEFAULT NULL,
```

```
PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

## ПРИЛОЖЕНИЕ Б

### Классы рейтинга эффективности работы

```
class GeneralConfig
{
    /**
     * @var array - массив встроенных в moodle переменных (Singleton)
     */
    protected $page;

    /**
     * @var \core_renderer
     */
    protected $output;

    /**
     * @var \moodle_database
     */
    protected $db;

    /**
     * @var \stdClass
     */
    protected $site;

    /**
     * @var \stdClass
     */
    protected $cfg;

    /**
     * @var \stdClass
     */
    protected $user;

    public function __construct()
    {
        $this->initMoodleVariable();
    }

    private function initMoodleVariable()
    {
        global $PAGE, $SITE, $OUTPUT, $DB, $CFG, $USER;

        $this->page = $PAGE;
        $this->site = $SITE;
        $this->output = $OUTPUT;
        $this->db = $DB;
        $this->cfg = $CFG;
        $this->user = $USER;
    }
}

<?php
class CGeneral extends GeneralConfig
{
    protected function render($template, array $params)
    {
```

```

        print $this->fetch($template, $params);
    }

    public function fetch($template, $params = array())
    {
        $template = $template . '.php';
        extract($params);
        if (file_exists($template))
        {
            ob_start();
            include $template;
            $result = ob_get_clean();
        } else
        {
            $result = 'Template file "' . $template . '" does not exists.';
        }

        return $result;
    }

    static public function staticFetch($template, $params = array())
    {
        $templater = new self();
        $result = $templater->fetch($template, $params);

        return $result;
    }

    public function escape($string)
    {
        return htmlspecialchars($string, ENT_QUOTES);
    }

    /**
     * This method add element in the end of $array
     * @param array $array
     * @param $key
     * @param $value
     * @return mixed
     */
    public function arrayPushAssoc($array, $key, $value)
    {
        $array[$key] = $value;
        return $array;
    }
}

class Route
{
    public static function Run($cmd, $do, $arg = '')
    {
        if (class_exists($cmd))
        {
            {
                $cmd = new $cmd();
                if (method_exists($cmd, $do))
                {
                    $cmd->$do($arg);
                    die();
                } else
                {
                    echo 'method<b>' . $do . '</b> is not exist';
                }
            }
        } else
    }
}

```

```

        {
            echo 'Class <b>' . $cmd . '</b> is not exist';
        }
    }
}

class ControllerPortfolio extends CGeneral
{
    private $modelPortfolio;
    private $accessUser;
    private $paramView = array();

    function __construct()
    {
        parent::__construct();
        $this->accessUser = new AccessUser();
        $this->modelPortfolio = new ModelPortfolio();
        $this->setTabCategory();
        $this->setCurrentPeriod();
    }

    /**
     * This method set tabs of categories
     */
    protected function setTabCategory()
    {
        $accessCategoryTab = $this->accessUser->getAccessCategoryTab();
        if (Check::isExistCatagory($accessCategoryTab))
        {
            $arrayTabsCategory = $this->modelPortfolio->getTabsCategory($accessCategoryTab);
            $this->paramView = array('arrayTabsCategory' =>
            $arrayTabsCategory);
        } else
        {
            $this->render('View/error/errorCategoryNotFound', array('error' =>
            'Нет доступных категорий'));
            die();
        }
    }

    /**
     * This method set current period of rating
     */
    protected function setCurrentPeriod()
    {
        $currentPeriod = $this->modelPortfolio->getCurrentPeriod(1);
        $this->paramView = $this->arrayPushAssoc($this->paramView,
        'nameperiod', $currentPeriod->nameperiod);
        $this->paramView = $this->arrayPushAssoc($this->paramView,
        'startperiod', $currentPeriod->startperiod);
        $this->paramView = $this->arrayPushAssoc($this->paramView, 'endperiod',
        $currentPeriod->endperiod);
    }

    /**
     * This method set all access tabs for portfolio by $categoryId
     * @param int $categoryId id category
     */
    public function setTabPortfolio($categoryId)
    {
        $accessPortfolioTab = $this->accessUser->

```

```

>getAccessPortfolioTab($categoryId);

        if (Check::isExistAccessPortfolioTab($accessPortfolioTab))
        {
            $arrayTabsPortfolio = $this->modelPortfolio-
>getTabsPortfolio($accessPortfolioTab);
            OutputPortfolio::printTabs($arrayTabsPortfolio, 'TabPortfolio');
        } else
        {
            echo '<b>Нет доступных вкладок портфолио</b><br>';
        }
    }

    /**
     * This method set singleSelect for period
     * @param int $categoryId id category
     */
    public function setSingleSelectPeriod($categoryId)
    {
        $arrayPeriod = $this->modelPortfolio-
>getPeriodForCategory($categoryId);

        OutputPortfolio::printSingleSelectPeriod($arrayPeriod, 'SingleSelectPeriod');
    }

    /**
     * This method make main page portfolio
     */
    public function index()
    {
        $this->render('View/viewPortfolio', $this->paramView);
    }
}

class ModelPortfolio extends GeneralConfig
{

    function __construct()
    {
        parent::__construct();
    }

    /**
     * This method return array of moodle's object tabobject
     * @param array $arrayCategory consists of a category number as a key and
value, if the value is 1,
     * then this category is available to the user. Otherwise 0
     * @return array $arrayTabsCategory
     */
    public function getTabsCategory($arrayCategory)
    {
        $category = "";
        foreach ($arrayCategory as $key => $value)
        {
            if ($value == 1)
            {
                $category .= $key . ', ';
            }
        }
        $category = substr($category, 0, -2);

        $query = "SELECT

```

```

        rcat.id,
        rcat.name
    FROM dean.mdl_bsu_rating_category rcat
    WHERE rcat.id in ($category)
    GROUP BY rcat.id";

    $resTmp = $this->db->get_records_sql($query);

    if (count($resTmp) == 0)
        echo '<p>Нет доступных категорий</p>';

    $tabs = array();

    foreach ($resTmp as $value)
    {
        $tabs[] = $value->name;
        $categories[] = $value->id;
    }

    $toprow = array();
    for ($i = 0; $i < count($tabs); $i++)
    {
        $arrayTabsCategory[] = new Tab('tabCategory' . $categories[$i],
        $categories[$i], $tabs[$i]);
    }

    return $arrayTabsCategory;
}

/**
 * This method return current period of rating by category
 * @param int $category
 * @return array $result
 */
public function getCurrentPeriod($category)
{
    $curDate = (date("Y-m-d"));

    $currentPeriodSql = "SELECT id
        FROM dean.mdl_bsu_rating_period
        where '$curDate' >= startperiod and '$curDate' <= endperiod and
categoryid = $category";

    if ($currentPeriod = $this->db->get_record_sql($currentPeriodSql,
null))
    {
        $result = $this->db->get_record_sql("SELECT
            nameperiod,
            DATE_FORMAT(startperiod,
'%d.%m.%Y') AS startperiod,
            DATE_FORMAT(endperiod,
'%d.%m.%Y') AS endperiod
        FROM mdl_bsu_rating_period
        WHERE id = $currentPeriod-
>id");
    } else
    {
        $result = "Не найден текущий период";
    }

    return $result;
}

```

```

/**
 * This method create tabs of objects portfolio
 * @param $arrayCategory
 * @return array
 */
public function getTabsPortfolio($arrayCategory)
{
    foreach ($arrayCategory as $category)
    {
        foreach ($category as $nameTab => $value)
        {
            if ($nameTab == 'mywork' and $value == true)
            {
                $arrayResult[] = new Tab('myworkPortfolio', 'mywork', "Мои
работы");
            } else if ($nameTab == 'view' and $value == true)
            {
                $arrayResult[] = new Tab("viewPortfolio", 'view',
"Просмотр");
            } else if ($nameTab == 'confirm' and $value == true)
            {
                $arrayResult[] = new Tab("confirmPortfolio", 'confirm',
"Подтверждение");
            }
        }
    }

    return $arrayResult;
}

/**
 * This method return all periods for $categoryId
 * @param int $categoryId
 * @return array $result
 */
public function getPeriodForCategory($categoryId)
{
    $periodByCategorySql = "SELECT
                            rp.id,
                            rp.nameperiod
                            FROM dean.mdl_bsu_rating_period rp
                            INNER JOIN mdl_bsu_rating_category rcat ON
rcat.id = rp.categoryid AND rcat.id = $categoryId";

    $menu = $this->db->get_records_sql_menu($periodByCategorySql);
    return $menu;
}

}

class Check
{
    /**
     * This method return true if exist available category. Otherwise false.
     * @param $array
     * @return bool
     */
    public static function isExistCatagory($arrayCategory)
    {
        $category = "";
        if (is_array($arrayCategory))
        {
            foreach ($arrayCategory as $key => $value)

```

```

        {
            if ($value == 1)
            {
                $category .= $key . ', ';
            }
        }
        $category = substr($category, 0, -2);
    }

    return ($category == "") ? false : true;
}

/**
 * This This method return true if exist available tabs's portfolio.
 * Otherwise false.
 * @param $arrayAccessPortfolio
 */
public static function isExistAccessPortfolioTab($arrayAccessPortfolio)
{
    $result = false;
    if (is_array($arrayAccessPortfolio))
    {
        foreach ($arrayAccessPortfolio as $category)
        {
            foreach ($category as $tab)
            {
                if ($tab == 1)
                {
                    $result = true;
                }
            }
        }
    }
    return $result;
}
}

```

## ПРИЛОЖЕНИЕ В

### Скрипт формирующий иерархическое дерево критериев рейтинга

```
<?php
require_once("../../config.php");
require_once("../../lib_blocks.php");
require_once("../../bsu_rating/portfolio/lib_listbox.php");
require_once("../../bsu_rating/role/ratingroles.php");
require_once("../../bsu_rating/lib_rating.php");
require_once("../../bsu_rating/lib_calcfunc.php");

require_login();

$kindworkingId = optional_param('kindworkingId', 0, PARAM_INT);
$categoryId = optional_param('category', 0, PARAM_INT);
$userid = optional_param('userid', $USER->id, PARAM_INT);
$subdep = optional_param('subdep', 0, PARAM_INT);
$departmentntcode = optional_param('departmentcode', 0, PARAM_INT);
$access = optional_param('access', "", PARAM_TEXT);
$periodId = optional_param('period', 0, PARAM_INT);
// $iscontroler - 0 у текущего пользователя нету подчиненных, 1 наоборот
// $iscontroler = optional_param('iscontroler', 0, PARAM_INT);
$waydisplay = optional_param('waydisplay', 0, PARAM_INT);
// передача userid обработчику
//require_once("../../bsu_rating/portfolio/handlerTree.php");

global $DB, $OUTPUT, $PAGE, $SITE, $CFG;

$scriptname = basename($_SERVER['PHP_SELF']);

$PAGE->requires->js('/lib/jquery/js/jquery-1.9.1.js', true);
$PAGE->requires->js('/lib/jquery/plugins/chosen/chosen.jquery.js', true);

$PAGE->requires->js('/blocks/bsu_rating/portfolio/treeWrapper.js', true);
$PAGE->requires->js('/blocks/bsu_rating/portfolio/treeBTN.js', true);

$strtitle = get_string('pluginname', 'block_bsu_rating');
$strtitle2 = get_string('portfolio', 'block_bsu_rating');
$strtitle3 = get_string('add_work', 'block_bsu_rating');

$PAGE->set_url('/blocks/bsu_ratrng/portfolio/tree.php');
$PAGE->set_context(context_system::instance());
$PAGE->set_title($strtitle3);
$PAGE->set_heading($SITE->fullname);
$PAGE->set_focuscontrol("");
$PAGE->navbar->add($strtitle, new moodle_url($CFG->wwwroot . '/blocks/bsu_rating/index.php', array()));
$PAGE->navbar->add($strtitle2, new moodle_url($CFG->wwwroot .
"/blocks/bsu_rating/portfolio/index.php?userid=$userid&category=$categoryId&period=$periodId", array()));
$PAGE->navbar->add($strtitle3);

echo $OUTPUT->header($userid);

$categoryName = get_value_from_ref("mdl_bsu_rating_category", "name", "id", $categoryId);
$categoryTable = get_value_from_ref("mdl_bsu_rating_category", "tablename", "id", $categoryId);

switch ($categoryTable) {
    case "mdl_bsu_ref_department": $categoryMember =
get_value_from_ref("$categoryTable", "name", "departmentcode", $departmentntcode); break;
    case "mdl_bsu_vw_ref_subdepartments": $categoryMember =
```

```

get_value_from_ref("$categoryTable","name","id",$subdep); break;
    case "mdl_bsu_ref_scientificunit": $categoryMember =
get_value_from_ref("$categoryTable","name","id",$subdep); break;
    case "mdl_user": $categoryMember = getuserNameById($userid); break;
}

// вычисление текущего периода
$currentYearId = get_current_edyearid();
switch($categoryId){
    case 3: {
        $role = new RatingRoles(0,1031, $subdep);
    }break;
    default: {
        $role = new RatingRoles(0,0, $subdep);
    }
}

global $allCriteriaIdWithAccess;
global $senderWorkWithSubdep;

$datePeriod = $DB->get_record_sql("SELECT startperiod, endperiodconfirm FROM mdl_bsu_rating_period
    WHERE id = $periodId");

$senderWorkWithSubdepSQL = " SELECT
        ew.id,
        rc.name,
        ew.criteriaid,
        ew.subdepid
    FROM mdl_bsu_rating_criteria AS rc
    INNER JOIN mdl_bsu_rating_enter_work AS ew ON rc.id = ew.criteriaid
    INNER JOIN mdl_bsu_rating_mark_criteria AS mc ON mc.criteriaid = rc.id
    WHERE ew.subdepid IS NOT NULL
        AND mc.onlyone IN (1)
        AND ew.dateconfirm BETWEEN '$datePeriod->startperiod' AND '$datePeriod-
>endperiodconfirm'";
//print_object($senderWorkWithSubdepSQL);

$senderWorkWithSubdep = $DB->get_records_sql($senderWorkWithSubdepSQL);

if (is_siteadmin()){
    // Получаем все допустимые id критериев
    $criteriaIds = $role->getAllCriteria($periodId, $categoryId, false);
    $allCriteriaIdWithAccess = $criteriaIds;
    $categoryStr = $role->getAllCriteria($periodId, $categoryId);
} else {
    $criteriaIds = $role->getCriteriaWithAccess($periodId, $categoryId, $access, false);
    $allCriteriaIdWithAccess = $criteriaIds;
    // exit();
    // print_object($allCriteriaIdWithAccess);

    $category = $role->getCriteriaWithAccess($periodId, $categoryId, $access);
    // echo $access;
    $roleids = $DB->get_record_sql("SELECT group_concat(distinct(roleid)) as grouprole FROM
    dean.mdl_role_assignments where userid=$userid");

    // print_object($roleids);

    // $criteriaIdsByRole = $role->getAccessCriteriaByRole("$roleids->grouprole",$periodId, $categoryId, $access,
false);

```

```

// print_object($criteriaIdsByRole);

// Дочерние критерии
$categoryStr = array();
foreach ($category as $key => $index) {
    $categoryStr[$key] = $key;
}
}
///

if (!empty($criteriaIds)) {
    $str_crit = "0";
    foreach($criteriaIds as $index => $criteriaId) {
        $str_crit .= "," . $index;
    }
    $criteriaIds = $str_crit;
} else {
    $criteriaIds = 0;
}

/*
echo "<p class='text-center'><b>Категория критериев рейтинга: </b>" . $categoryName . "</p>";

// Текст - "от имени : ФИО", значение берется из таблицы mdl_user по $userid
echo "<p class='text-center'><b>Участник рейтинга: </b>" . userNameById($userid) . "</p>";
*/

// формируем низпадающий список видов деятельности
echo '<table class="generalbox" border=1 cellspacing="0" cellpadding="0" align="center">'.
    '<tr align="left"><td align=right style="font-weight: bold">Категория критериев рейтинга: </td><td
align="left">'.
    $categoryName.
    '</td></tr>'.
    '<tr align="left"><td align=right style="font-weight: bold">Участник рейтинга: </td><td align="left">'.
    $categoryMember.
    '</td></tr>'.

listbox_reference("$scriptname?access=$access&category=$categoryId&userid=$userid&period=$periodId&iscon
troller=$iscontroller&subdep=$subdep", $kindworkingId, $criteriaIds).

/*listbox_waydisplay("$scriptname?access=$access&category=$categoryId&userid=$userid&period=$periodId&ki
ndworkingId=$kindworkingId&iscontroller=$iscontroller", $waydisplay).*/</table>';
if ($kindworkingId > -1) {
    // Условие выбора критериев

    if ($kindworkingId == 0){
        // Условие выбора критериев
        $condition = "";
    } else {
        $condition = "and rc.kindworkid in ($kindworkingId)";
    }

    $DB->Execute("create temporary table t1
        SELECT rc.id, rc.categoryid, rc.kindworkid, rc.parentid, rc.name, rc.type_enter, rc.path,
        rc.datein, rc.oidid, rc.oldparentid, rmc.id as rmc_ID, rmc.periodid, rmc.criteriaid, rmc.mark,
        rmc.sortnumber, rmc.calcfunc, rmc.onlyone, rk.name as nameKindwork, rk.contact
        FROM dean.mdl_bsu_rating_criteria as rc inner join mdl_bsu_rating_mark_criteria as rmc on rc.id
= rmc.criteriaid
        inner join mdl_bsu_rating_kindworking as rk on rc.kindworkid = rk.id
        where rc.id in ($criteriaIds) and rmc.periodid = $periodId $condition order by sortnumber");

    $allCriteriaes = $DB->get_records_sql("select * from t1 order by sortnumber");

```

```

foreach ($allCriteria as $criteria) {
    // if(is_siteadmin()) echo $criteria.'<br>';
    $arrCategories[$criteria->parentid][$criteria->id] = $criteria;
}

global $temp_tree;
global $flag;
global $PeriodIdForTree;
$PeriodIdForTree = $periodId;

$flag = 0;

// print_object($arrCategories);
$tree = build_tree($arrCategories, 0, $categoryStr,$skindworkingId, $subdep);
$btn = "<input class=\"buttonNext\" type=\"button\" value=\"Далее >> \"
onclick=\"check(\$userid,$categoryId, \$periodId, \$departmentntcode, \$subdep)\"/>";
echo $btn;
echo '<div class="menuTree">'.
    '<p>'.
        '<a title="раскрыть все записи">
            <label id="waydisplay">."раскрыть все записи".
        </label>'.
        </a>'.
        </span>'.</p>'.
    </div>';

echo '<div class="listContainer">' . $tree . '</div>';
echo $btn;
// }
}

echo $OUTPUT->footer();
// формируем иерархическое дерево

function build_tree($criteria_array,$parent_id,$criteriaNotParent, $skindworkId, $subdep=0){

    /*
    * Записываем текущий период в глобальную переменную $PeriodIdForTree
    * для дальнейшего использования в функции build_tree
    */
    global $DB;

    if(is_array($criteria_array) and isset($criteria_array[$parent_id])){
        $tree = '<ul>';
        foreach($criteria_array[$parent_id] as $criteria){
            /*
            * при первой итерации цикла, $GLOBALS['$flag'] равен 0 во всех остальных случаях он равен 1;
            * при первой итерации цикла, $GLOBALS['temp_tree'] присваивается значение namekindwork;
            * во всех остальных итерациях цикла происходит сравнение $GLOBALS['temp_tree'] с текущим
            критерием $criteria->namekindwork,
            * если они не равны то выводим $criteria->namekindwork;
            */
            if($GLOBALS['flag'] == 0){
                $GLOBALS['temp_tree'] = $criteria->namekindwork;
                $GLOBALS['flag'] = 1;
                $tree .= '<p>'.$criteria->namekindwork." ($criteria->contact)".</p>';
            }else if(($GLOBALS['flag'] > 0) && strcmp($GLOBALS['temp_tree'], $criteria->namekindwork) != 0){
                $GLOBALS['temp_tree'] = $criteria->namekindwork;
                $tree .= '<p>'.$criteria->namekindwork." ($criteria->contact)".</p>';
            }
        }
        if (in_array($criteria->id,$criteriaNotParent)){

```





```

return $tree;
}

function findAccess($arrayCriteriaWithAccess, $criteriaId){
    if(isset($arrayCriteriaWithAccess[$criteriaId])){
        return $arrayCriteriaWithAccess[$criteriaId];

    } else {
        return null;
    }
}
/*
 * @arrayEnterWorkCriteriaWithSubdep - массив объектов содержащий id критериев и id подразделений
 * из таблицы mdl_bsu_rating_enter_work
 * @criteriaId - id критерия
 * @subdepId - id подразделения
 * @return 0 - Данный критерий заполнен
 *         1 - Данный критерий не заполнен
 *         3 - Данный критерий не уникальный, следовательно его можно заполнять несколько раз
 */
function checkUniqueCriteria($arrayEnterWorkCriteriaWithSubdep,$criteriaId, $subdepId){
    foreach ($arrayEnterWorkCriteriaWithSubdep as $criteria) {
        if($criteria->criteriaid == $criteriaId){
            if ($criteria->subdepid == $subdepId) {
                return 0;
            }else{
                $ret = 1;
            }
        }
    }
}

if (isset($ret) && $ret == 1){
    return $ret;
}else{
    return 3;
}
}
?>

```

## ПРИЛОЖЕНИЕ Г

```
$(function () {
    var action = $('#switchTableName').find('select').val();
    switch (action) {
        case '1': {
            var urlString = "bsu_rating_category";
            var caption = "Просмотр и редактирование категорий рейтинга";
            initJQGridCategory(urlString, caption);
        }
        break;
        case '2': {
            var urlString = "bsu_rating_kindworking";
            var caption = "Просмотр и редактирование видов деятельности рейтинга";
            initJQGrid(urlString, caption);
        }
        break;
        case '3': {
            var urlString = "bsu_rating_status_inputdata";
            var caption = "Просмотр и редактирование статусов введенных работ рейтинга";
            initJQGrid(urlString, caption);
        }
        break;
        case '4': {
            var yearId = $('#switchYearId').find('select').val();
            document.getElementById('selectYearRating').style.display = "block";
            var urlString = "bsu_rating_period" + "&yearId=" + yearId;
            var caption = "Просмотр и редактирование категорий рейтинга";
            initJQGridPeriod(urlString, caption);
        }
        break;
        case '5': {
            var criteriaId = $('#switchCriteriaId').find('select').val();
            document.getElementById('selectCriteriaRating').style.display = "block";
            document.getElementById('selectKindworkingRating').style.display = "block";
            document.getElementById('selectPeriodRating').style.display = "block";
            document.getElementById('selectCategoryRating').style.display = "block";
            var urlStringWithCriteria = "bsu_rating_fielddata&criteriaId=" + criteriaId;
            var captionDistinctFieldData = "Ранне созданные поля";
            var urlString = "bsu_rating_fielddata" + "&criteriaId=" + criteriaId;
            var captionFieldData = "Поля работы";
            if (criteriaId > 0) {
                document.getElementById('tabs').style.display = "block";
                var getParametr = parseGetParams();
                if (getParametr['tab'] == 1) {
                    document.getElementById('middle').style.display = "block";
                    initJQGridRatingFieldData(urlString, captionFieldData);
                    initJQGridRatingDistinctFieldData(urlStringWithCriteria, captionDistinctFieldData);
                    $("#btnAdd").click(function () {
                        var s;
                        s = $("#tableRight").jqGrid('getGridParam', 'selarrrow') + '|' + getParametr['criteriaId'];
                        console.log(s);
                        $.ajax({
                            type: "POST",
                            url: "addRatingFieldForCriteria.php",
                            data: "text=" + s,
                            success: function () {
                                jQuery("#tableLeft").jqGrid('setGridParam', {datatype: 'xml'}).trigger('reloadGrid');
                            }
                        });
                    });
                }
            }
        }
    }
});
```

```

    } else {
        AutocompleteName();
        AutocompleteFieldName();
        // AutocompleteFieldType();
        $(document).ready(function () {
            $("#btn_add_new_field_id").click(
                function () {
                    jQuery("#table").trigger("reloadGrid");
                    sendAjaxForm('form_fields', 'saveNewField.php?criteriaId=' + getParametr['criteriaId']);
                    return false;
                }
            );
        });
    }
}
break;
}
case '6': {
    var getParametr = parseGetParams();
    var categoryId = $("#switchCategoryId").find('select').val();
    var kindworkid = $("#switchedkindwork").find('select').val();
    var periodId = $("#switchPeriodId").find('select').val();
    document.getElementById('selectCategoryRating').style.display = "block";
    document.getElementById('selectKindworkingRating').style.display = "block";
    document.getElementById('selectPeriodRating').style.display = "block";
    var urlString = "bsu_rating_criteria";
    var caption = "Просмотр и редактирование критериев рейтинга";
    if (categoryId != 0) {
        var template = "";
        initJQGridRatingCriteria(urlString, caption, categoryId, kindworkid, periodId, template);
    } else {
        $("#error").text("Выберите категорию");
    }
}
break;
}
});
/**
 * Данная функция создает таблицу с двумя столбцами
 * @param string $urlString название таблицы откуда берутся данные
 * @param string $caption название таблицы
 */
function initJQGrid(urlString, caption) {
    jQuery("#table").jqGrid({
        url: 'data.php?tablename=' + urlString,
        datatype: "xml",
        mtype: "GET",
        colNames: ["id", "Наименование"],
        colModel: [
            {
                name: "id",
                width: 10,
                editable: false,
            },
            {
                name: "name",
                width: 80,
                editable: true,
                editoptions: {
                    size: 60
                }
            }
        ]
    });
}

```

```

    },
  ],
  editurl: 'save.php?tablename=' + urlString,
  pager: "#pager",
  rowNum: 20,
  rowList: [10, 20, 30],
  sortname: "id",
  sortorder: "asc",
  viewrecords: true,
  gridview: true,
  autoencode: true,
  altrows: true,
  caption: caption,
  height: 470,
  width: 1200,
});

$("#table").jqGrid('navGrid', '#pager',
  {edit: true, view: false, add: true, del: true}, //options
  {left: 200, top: 250, height: 140, width: 600, reloadAfterSubmit: true}, // edit options
  {height: 140, width: 600, reloadAfterSubmit: true}, // add options
  {reloadAfterSubmit: true}, // del options
  {top: 400, left: 300, height: 200, width: 600, reloadAfterSubmit: false} // search options
);

```



Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«\_\_\_» \_\_\_\_\_ Г.

---

*(подпись)*

---

*(Ф.И.О.)*