Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский**

**технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | И.А. Серышева |
|  | И.О. Фамилия |

|  |
| --- |
|  |
| Основные проектные решения по разработке Веб-сервиса автоматизации «Системы эффективного контракта» для ИЮИ (ф) УП РФ |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

|  |
| --- |
| Проектирование информационных систем |

|  |
| --- |
| 1.033.00.00 ПЗ |
| обозначение документа |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | ИСТб-21-1 |  |  | |  | Е.А. Юрковский |
|  |  | шифр группы |  | подпись | |  | И.О. Фамилия |
| Нормоконтроль |  |  |  |  | |  | И.А. Серышева |
|  |  |  |  | подпись | |  | И.О. Фамилия |
| Курсовой проект защищен с оценкой | | | | |  | | | |

Иркутск 2025 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

ЗАДАНИЕ

на курсовое проектирование

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| По курсу | Проектирование информационных систем | | | | | | | | | | | | | | | | | | | | |
| Студенту | Юрковскому Е.А. | | | | | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы) | | | | | | | | | | | | | | | | | | | | |
| Тема проекта | | Основные проектные решения по разработке Веб-сервиса автоматизации «Системы эффективного контракта» для ИЮИ (ф) УП РФ | | | | | | | | | | | | | | | | | | | |
| Исходные данные | | | Разработать фулстэк веб-сервис, которая автоматизирует процесс формирования рейтинга сотрудников ИЮИ (ф) УП РФ в рамках «Системы эффективного контракта». | | | | | | | | | | | | | | | | | | |
| Рекомендуемая литература | | | | | | | |  | | | | | | | | | | | | | |
| 1. Проектирование АСОИУ : [ Электронный ресурс] : методические указания по выполнению курсового проекта: 09.03.01 «Информатика и вычислительная техника» для бакалавров по специальности «Автоматизированные системы обработки информации и управления» / Иркут. нац. исслед. техн. ун-т ; сост. Р. Д. Гутгарц. - Иркутск : ИРНИТУ, 2018. - 13 с. - URL: http://elib.istu.edu/viewer/view.php?file=/files3/er-17727.pdf. | | | | | | | | | | | | | | | | | | | | | |
| Графическая часть на | | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ листах. | | | | | | | | | | | | | | | | | |
| Дата выдачи задания | | | | « | 26 | » | сентября | | | | | | 2024 г. | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | |
| Задание получил | | | | | | | | |  | | | | | |  | | Е.А. Юрковский | | | | |
|  | | | | | | | | | подпись | | | | | |  | | И.О. Фамилия | | | | |
|  | | | | | | | | | | |  |  | |  | |  | | | |  | |
| Дата представления проекта руководителю | | | | | | | | | | | « | 25 | | » | | февраля | | | | 2025 г. | |
|  | | | | | | | | | | | | | | | | | | | | | |
| Руководитель курсового проектирования | | | | | | | | | |  | | | | | | | |  | И.А. Серышева | | |
|  | | | | | | | | | | подпись | | | | | | | |  | И.О. Фамилия | | |

# Термины и сокращения предметной области

В настоящем курсовом проекте применятся термины и сокращения с соответствующими определениями, представленные в таблице 1.

Таблица 1 – Список терминов и сокращений

|  |  |
| --- | --- |
| Термин | Определение |
| ИС | Информационная система |
| ИЮИ (ф) УП РФ | Иркутский юридический институт филиал университета прокуратуры. |
| KPI | Key performance index (Ключевые показатели эффективности) |
| Система эффективного контракта | Форма трудового договора, в котором указаны показатели эффективности деятельности работника, трудовые обязанности, исполнение которых необходимо для достижения установленных показателей эффективности |

**Содержание**

[Термины и сокращения предметной области 3](#_Toc192006033)

[Введение 6](#_Toc192006034)

[1 Предпроектное исследование 8](#_Toc192006035)

[1.1 Критический анализ и моделирование текущего процесса формирования рейтинга университета 8](#_Toc192006036)

[1.2 Обзор существующих аналогов ИС «Эффективного контракта» 14](#_Toc192006037)

[1.3 Постановка задачи 17](#_Toc192006038)

[1.4 Требования к разрабатываемому обеспечению 19](#_Toc192006039)

[1.4.1 Общие требования 19](#_Toc192006040)

[1.4.2 Функциональные требования 22](#_Toc192006041)

[Заключение по разделу 26](#_Toc192006042)

[2 Проектирование ИС 28](#_Toc192006043)

[2.1 Процесс формирования рейтинга с применением проектируемой ИС 28](#_Toc192006044)

[2.2 Схема потоков данных ИС 32](#_Toc192006045)

[2.3 Задача унификации показателей 36](#_Toc192006046)

[2.4 Схема данных ИС 37](#_Toc192006047)

[2.4.1. Инфологическая модель 37](#_Toc192006048)

[2.4.2. Физическая модель 38](#_Toc192006049)

[2.4.3. Модель БД в разрезе PostgreSQL модель 39](#_Toc192006050)

[2.5 Описание макетов интерфейса. 40](#_Toc192006051)

[2.6 Архитектура системы 46](#_Toc192006052)

[2.7 Инструменты разработки 51](#_Toc192006053)

[Заключение по разделу 53](#_Toc192006054)

[3 Реализация проекта 55](#_Toc192006055)

[3.1 Реализация базы данных 55](#_Toc192006056)

[3.2 Реализация API-запросов 57](#_Toc192006057)

[3.3 Отправка уведомлений на почту 60](#_Toc192006058)

[3.4 Описание клиентской части 62](#_Toc192006059)

[3.5 Модуль генерации отчетов 66](#_Toc192006060)

[3.6 Описание возможностей текущей версии 67](#_Toc192006061)

[3.7 Пример работы с приложением 68](#_Toc192006062)

[3.8 Возможное развитие системы 71](#_Toc192006063)

[Заключение по разделу 73](#_Toc192006064)

[4 Экономическая часть 74](#_Toc192006065)

[Заключение 78](#_Toc192006066)

[Список литературы 79](#_Toc192006067)

# Введение

Иркутский юридический институт (филиал) федерального государственного казенного образовательного учреждения высшего образования «Университет прокуратуры Российской Федерации» является образовательным заведением, которое обеспечивает качественную всестороннюю подготовку работников прокуратуры, не только со всей области, но и на федеральном и межгосударственном уровнях.

Поддержание высокого качества образования является приоритетной задачей сотрудников института. В современных условиях, в Российской Федерации, для достижения поставленной цели особое значение приобретает система оценки эффективности работы сотрудников образовательных учреждений. Одной из наиболее распространенных методик является «Система эффективного контракта», основанная на применении KPI (Key Performance Indicators). Именно она используется в данном учебном заведении.

Она направлена на стимулирование профессиональной деятельности преподавателей через привязку заработной платы к их достижениям и выполнению определенных показателей эффективности. Рейтинг преподавателей важен для поддержания высокого уровня образовательного процесса и стимулирования преподавателей к постоянному росту и улучшению своей работы. Результаты оценки могут использоваться для принятия решений о продлении контракта, повышении зарплаты, предоставлении академических наград, выделении ресурсов для дальнейшего развития.

Система KPI включает в себя множество критериев, которые формируются и определяются в документах от головного университета [2]. Полученные критерии всегда сопровождаются шаблонами, которые определяют способ контроля за соблюдением критериев, как индивидуально для сотрудника [3], так и для кафедры в целом [4].

Для крупных вузов, обладающих достаточным финансированием и кадровыми ресурсами, разработаны информационные системы (ИС), которые автоматизируют учет показателей и облегчают работу сотрудников. Такие программные комплексы позволяют преподавателям самостоятельно вводить данные о своих достижениях, после чего они автоматически агрегируют, анализируются и используются для расчета итогового балла KPI. Это сокращает временные затраты, снижает вероятность ошибок и упрощает документооборот.

Однако в небольших вузах, где финансирование ограничено, процессы сбора, обработки и анализа данных, до сих пор, зачастую, ведутся вручную или с использованием электронных таблиц, в частности - EXCEL. Такой подход имеет ряд недостатков:

* Высокая трудоемкость: сотрудники вынуждены тратить значительное время на внесение и обработку данных.
* Повторный ввод данных: одни и те же показатели часто используются в разных отчетах и документах, что увеличивает вероятность дублирования информации и возникновения ошибок.

Разработка автоматизированной информационной системы, которая будет способна автоматизировать процессы, связанные с формированием рейтинга сотрудников в рамках «Системы эффективного контракта» позволит значительно снизить нагрузку на сотрудников, ответственных за проведение данного процесса.

**Целью** выпускной квалификационной работыявляется снижение нагрузки на ответственных за процесс формирования рейтинга сотрудников университета ИЮИ (ф) УП РФ в рамках «Системы эффективного контракта» за счет внедрения ИС, которая будет способна на автоматический перерасчёт KPI сотрудников университета, централизованное хранение данных о списках достижений сотрудников, формирование итоговых рейтингов по университету, по подразделениям, внутри подразделений, а также генерацию отчётных документов, согласно шаблонам, что повысит точность, сократит время и упростит деятельность в периоды формирования рейтинга.

**Задачи**, которые необходимо выполнить, для успешного достижения целипредставленной выпускной квалификационной работы:

* проанализировать текущий процесс формирования рейтинга в ИЮИ (ф) УП РФ;
* спроектировать новый бизнес-процесс «Формирование рейтинга университета в рамках системы эффективного контракта» с применением разрабатываемой ИС;
* cпроектировать систему – архитектуру, базу данных, серверную и клиентскую части, интерфейс;
* реализовать спроектированную ИС;
* провести интеграцию и развёртывание ИС на мощностях ИЮИ (ф) УП РФ;
* организовать и провести обучение сотрудников университета.

Результатом данной работы станет автоматизированная информационная система, которая будет способна вести все расчеты, связанные с процессом формирования рейтинга, централизовать процесс, при помощи хранения всех данных системы в одном месте, а также иметь возможность генерировать всю необходимую отчетность, в значительной степени упростит процесс формирования рейтинга сотрудников университета и значительно снизит нагрузку участвующих в этом процессе лиц, особенно в периоды активного составления рейтинговых списков. Кроме того, система устранит человеческий фактор, путем отсутствия необходимости документооборота между этапами процесса создания рейтинга и позволит с легкостью получать связанные с рейтингом данные за прошлые периоды, без необходимости поднимать архивы бумаг.

# Предпроектное исследование

## Критический анализ и моделирование текущего процесса формирования рейтинга университета

Формирование рейтинга сотрудников целого университета — это трудоемкий процесс, который требует затрат большого количества сил и времени, особенно у сотрудников, ответственных за этот процесс. В ИЮИ (ф) УП РФ, штат сотрудников которого составляет около 100 человек, система «Эффективного контракта», которая применяется для многих ВУЗов страны, также используется в целях оценки эффективности работы преподавателей. Однако, в отличие от крупных учебных заведений, где существуют автоматизированные информационные системы, которые берут на себя всю механическую часть работы, весь процесс расчета показателей и подготовки отчетов осуществляется вручную, используя заранее предоставленные приказами шаблоны еxcel-документов[3,4, 5], которые определяют внешний вид, наполнение и процесс заполнения списков достижений сотрудников, из которых формируется KPI, а также ряд условий, для подтверждения того или иного достижения. Пара сотрудников из штата университета назначаются ответственными за этот рейтинг, рейтинг составляется раз в квартал. Назначенные ответственными заполняют нормативные шаблоны основываясь на предоставленной им информации, ищут и до запрашивают подтверждающие документы, при их отсутствии, а также регулируют, чтобы достижения соответствовали заявленным критериям и все условия начисления были соблюдены

Процесс заполнения данных о достижениях сотрудников ответственными за рейтинг происходит поэтапно:

1. На первом этапе составляется список достижений каждого сотрудника, в который вручную вносятся сведения о научных публикациях, участии в конференциях, преподавательской нагрузке, учебно-методической деятельности и других показателях [2]. При этом формулы, автоматизирующие табличный учет, в документах, как правило, отсутствуют, поэтому каждый расчет осуществляется вручную, что увеличивает вероятность ошибок;
2. На втором этапе результаты по баллам для каждого сотрудника дублируются в рейтинговую таблицу внутри подразделения. Это требует повторного ввода тех же данных, что отнимает дополнительное время и повышает вероятность допущения опечаток или иных неточностей;
3. На заключительном этапе данные снова переносятся вручную в сводную таблицу по всему университету, где формируется общий рейтинг преподавателей и рейтинг подразделений. Таким образом, одни и те же данные вводятся минимум трижды, что не только увеличивает трудоемкость, но и создает риск потери данных или возникновения несоответствий между разными таблицами.

В текущий момент процесс формирования рейтинга сотрудников университета состоит из следующих задач:

1. Заполнение списков достижений сотрудников;
2. Сведения достижений по каждому сотруднику, расчет общего KPI;
3. Контроль корректности внесенных достижений;
4. Сведения показателей каждого сотрудника в рейтинг по университету, заполнение списка итоговых баллов KPI (рейтинг сотрудников университета);
5. Сведения показателей каждого сотрудника в подразделении в рейтинг по подразделению, заполнение списка итоговых баллов KPI (рейтинг сотрудников подразделения);
6. Сведения рейтингов каждого подразделения в рейтинг подразделений с помощью расчета общего среднего балла KPI по каждому подразделению, заполнение списка итогов по подразделениям (рейтинг подразделений (кафедр) университета).

Каждый из этапов сопровождается бумажным документооборотом. Необходимо перенести бумаги от одного человека к другому для того, чтобы приступить к следующему этапу процесса и получить всех необходимых подписи. Также необходимо идти и спрашивать об отсутствующих подтверждающих документов и иногда даже брать в руки физические носители (Подтверждающие документы) для того, чтобы использовать его, как подтверждение достижения.

В общем, задача формирования рейтинга сводится к тому, чтобы **все имеющиеся достижения сотрудника, согласно показателям [2], были учтены и записаны**. После происходит пересчет итога по баллам, а также среднего бала по подразделению. Итогом всего процесса будут сформированные рейтинги, а также комплект отчетности по ним, который состоит из сводной таблицы по сотрудникам подразделения с проставленными значениями в показателях оценки [3], заполненной индивидуальной таблицы достижений по показателям сотрудника [4], а также сводные таблицы рейтингов сотрудников по университету и сводной таблицы рейтинга подразделений университета. Ниже на рисунке 1.1.1 представлена общая схема процесса «Формирование рейтинга сотрудников университета» на общем уровне, а также, на рисунке 1.1.2, представлена декомпозиция этого процесса, которая состоит из составления списка достижений, а также формирования каждого из типов рейтинга.

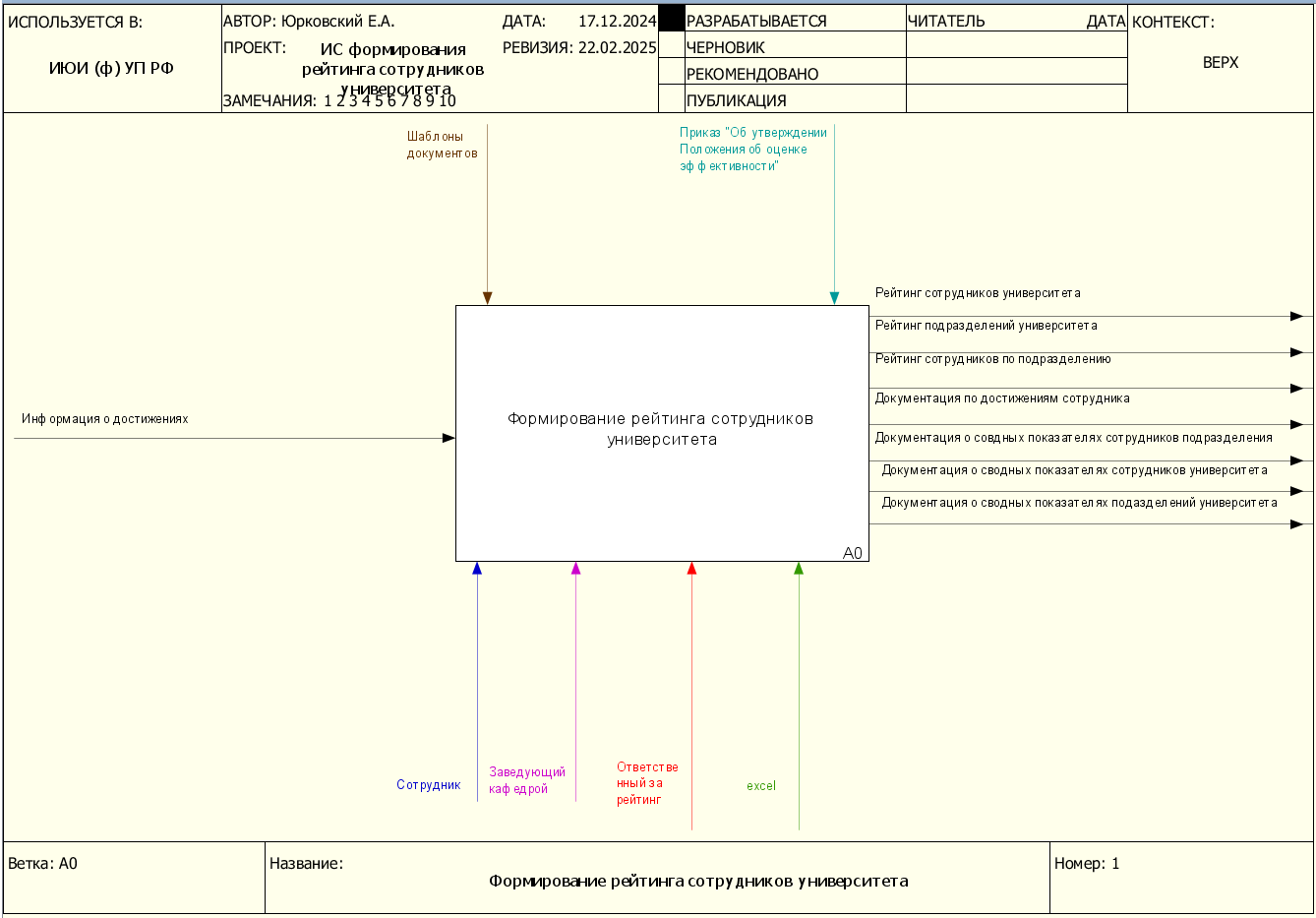


Рисунок 1.1.1 -Схема IDEF0 AS-IS. Уровень A-0. Формирование рейтинга сотрудников университета

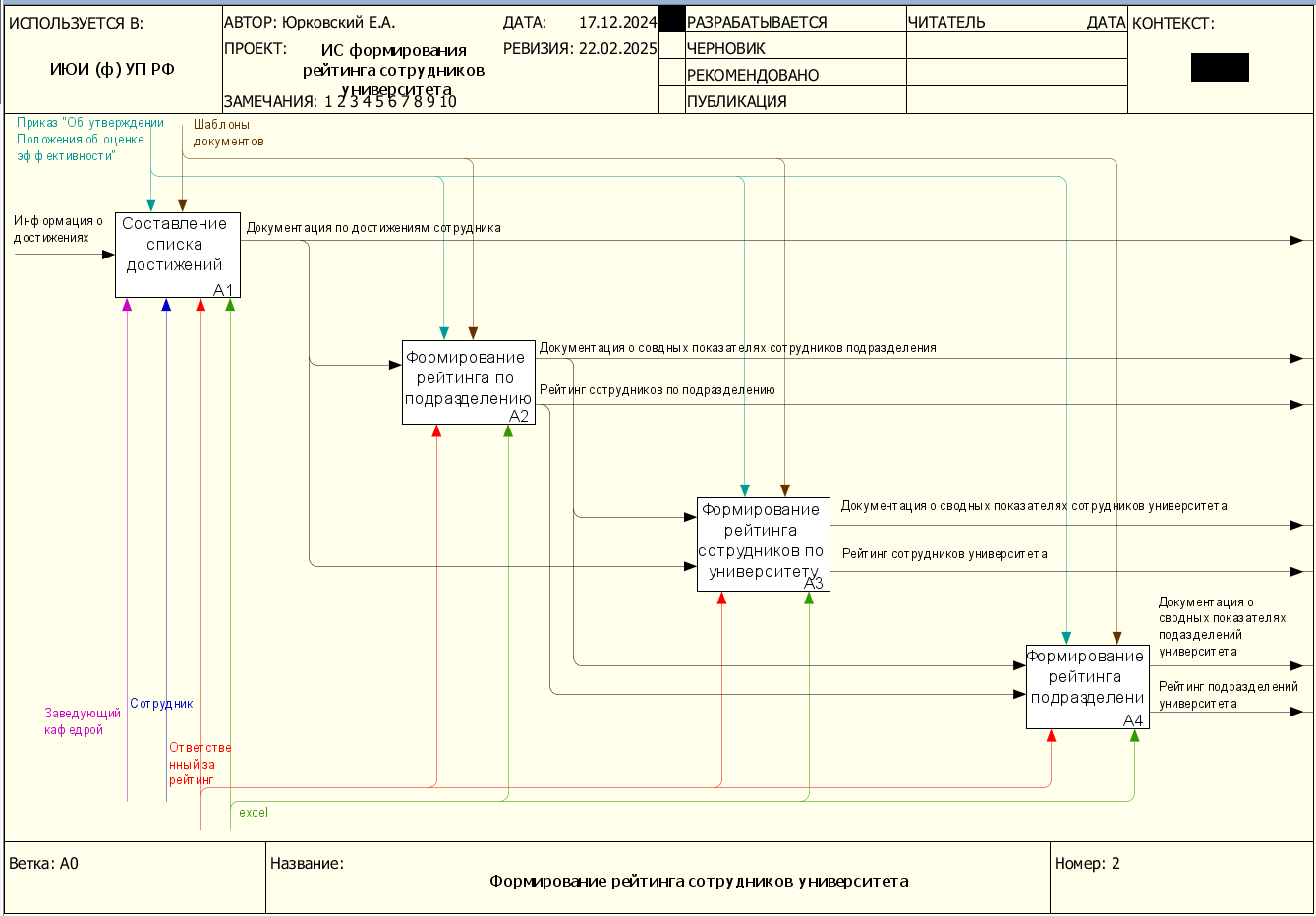


Рисунок 1.1.2 -Схема IDEF0 AS-IS. Уровень A0. Формирование рейтинга сотрудников университета

Основным процессом, вышеописанным как первый этап процесса, является составление списков достижений по каждому сотруднику, так как все последующие процессы, по своей сути, являются просто различными видами компоновки рассчитанных результатов на этом этапе. Для того, чтобы заполнить достижения, сотрудник должен внести по каждому показателю, который соответствует, имеющемуся достижению информацию, затем пересчитать все свои баллы в соответствии с показателями и подпунктами показателей [2], для каждого подпункта показателя, которые требуют специфический пересчет итога. После внесения всех имеющихся достижений необходимо рассчитать и внести общий балл (свой KPI), а также добавить к каждому из внесенных достижений подтверждающие документы / условия начисления достижений [2], для каждого подпункта показателя. После формирования своего списка достижений сотрудник направляет его на проверку «Ответственному за формирование рейтинга». Ниже, на рисунке 1.1.3 представлена схема декомпозиции процесса составление списка достижений сотрудника.

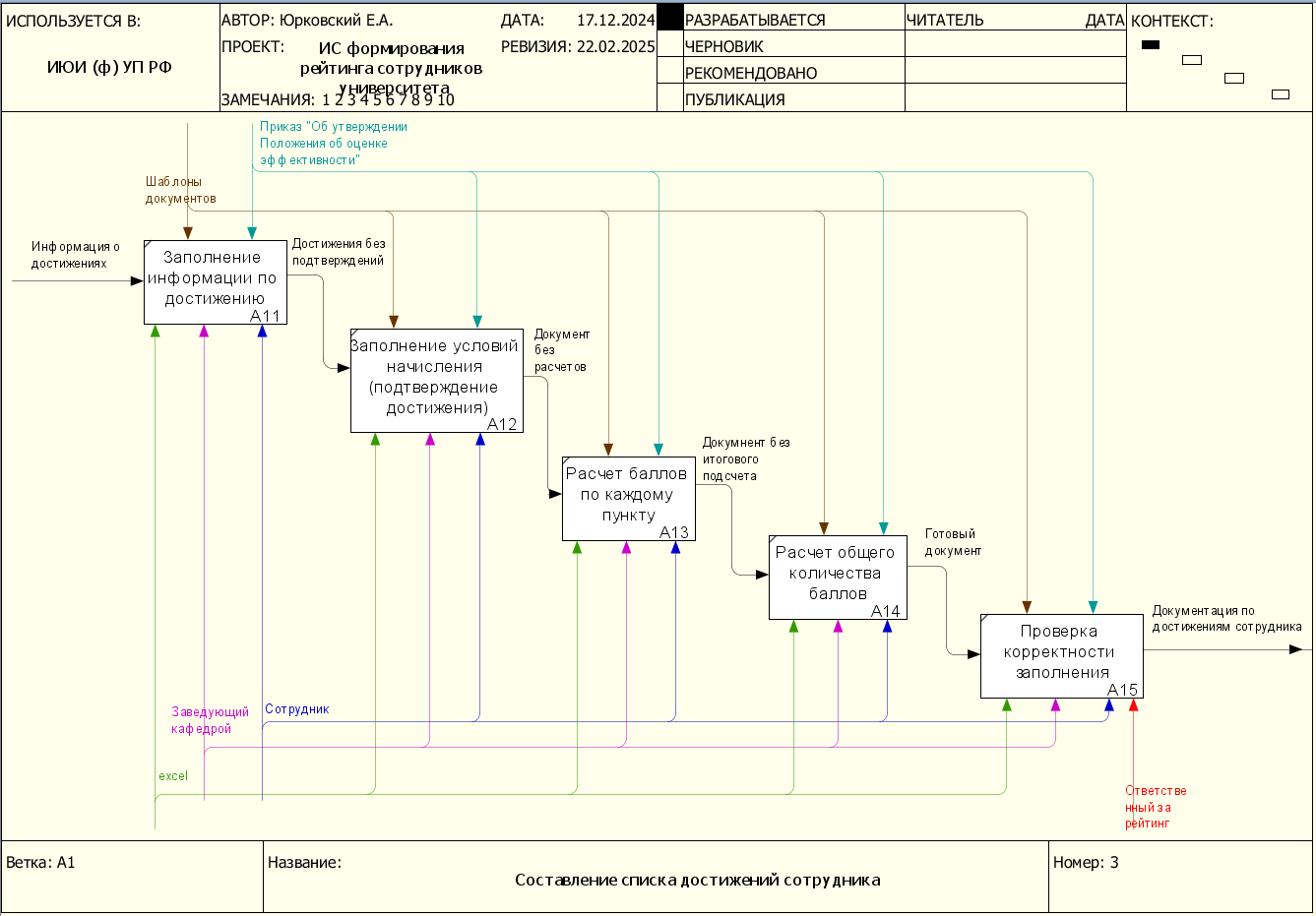


Рисунок 1.1.3 -Схема IDEF0 AS-IS. Уровень A1. Декомпозиция процесса составление списка достижений сотрудника

Все дальнейшие процессы формирования рейтинга(ов) являются, по большей части, различными способами компоновки имеющихся результатов достижений сотрудников. Процесс, описывании на рисунке 1.1.4 показывает процесс формирования рейтинга по подразделению, что является процессом сведения списков достижений сотрудников в одну общую таблицу, отсортированную по убыванию KPI сотрудника. На рисунке 1.1.5 тот же процесс, но теперь учитываются вообще все сотрудники. На рисунке 1.1.6 происходит тот же процесс, но также еще рассчитывается средний показатель KPI по подразделению для составления объективного рейтинга, в связи с тем, что в разных подразделениях состоит разное количество сотрудников, что может валится в очевидный дисбаланс баллов, если бы рейтинг по подразделениям формировался на основании общего бала подразделения. Ниже представлены рисунки 1.1.4, 1.1.5, 1.1.6, которые описывают процесс формирования рейтингов.

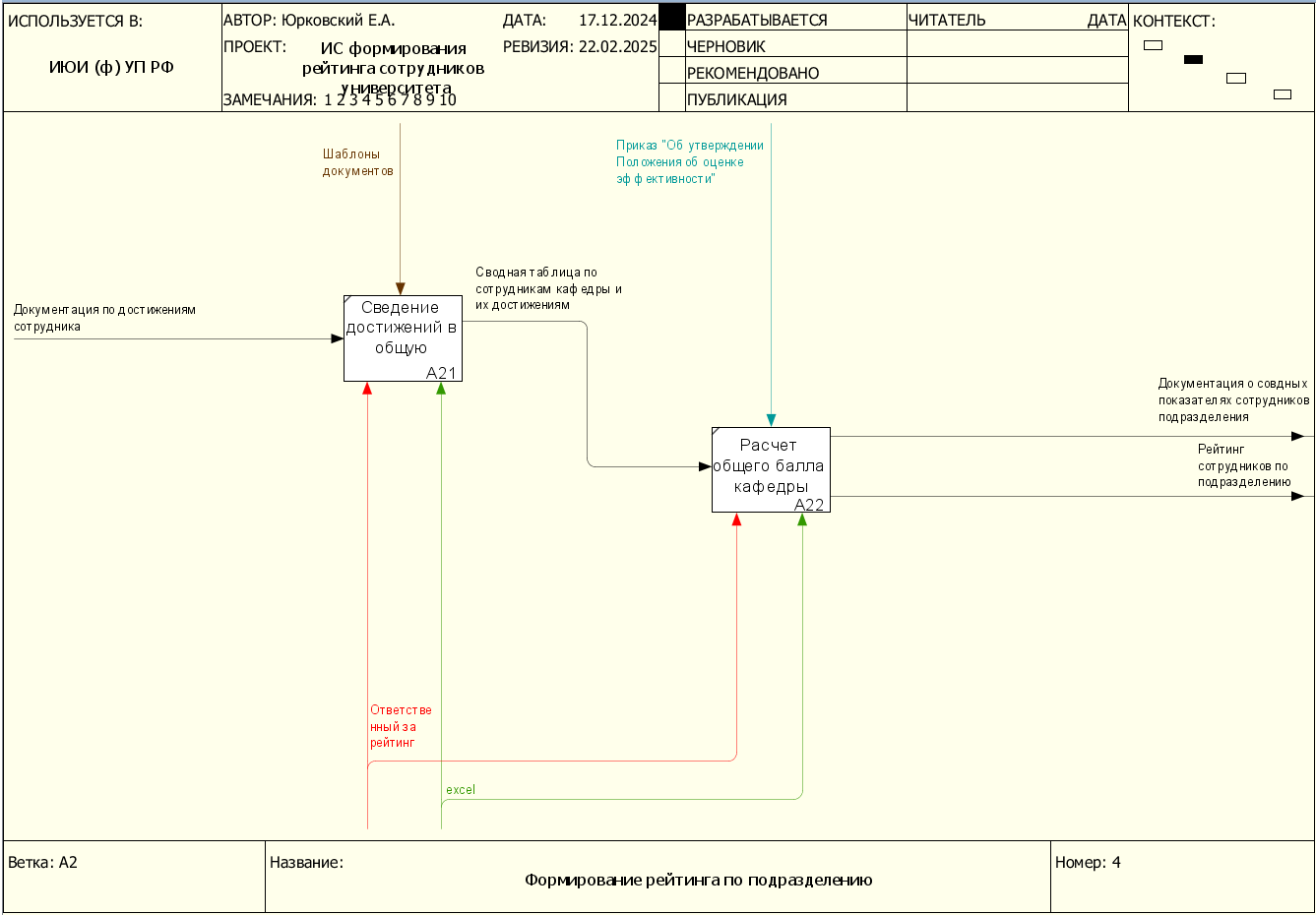


Рисунок 1.1.4 -Схема IDEF0 AS-IS. Уровень A2. Декомпозиция процесса формирования рейтинга по подразделению

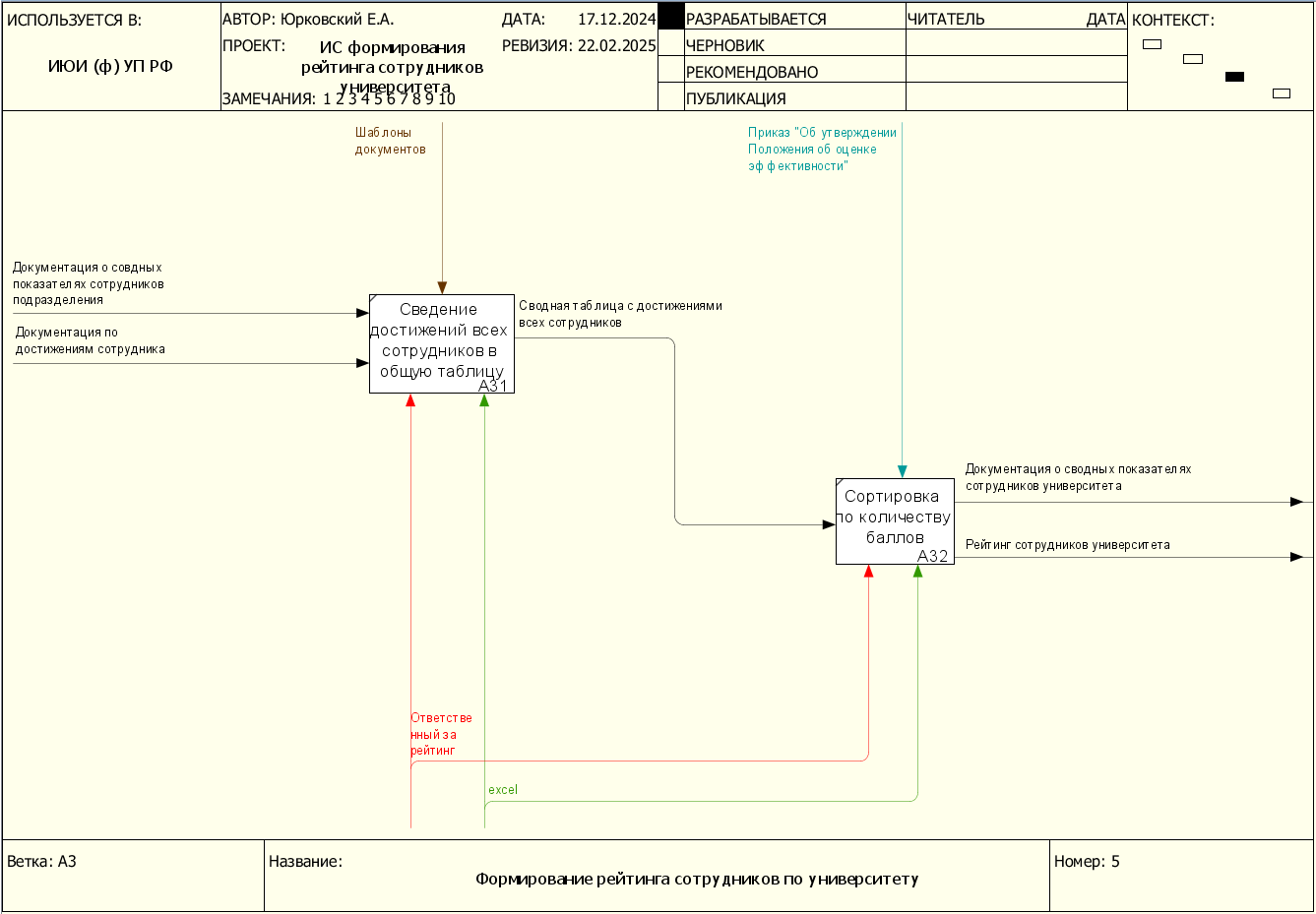


Рисунок 1.1.5 -Схема IDEF0 AS-IS. Уровень A3. Декомпозиция процесса формирование рейтинга сотрудников по университету

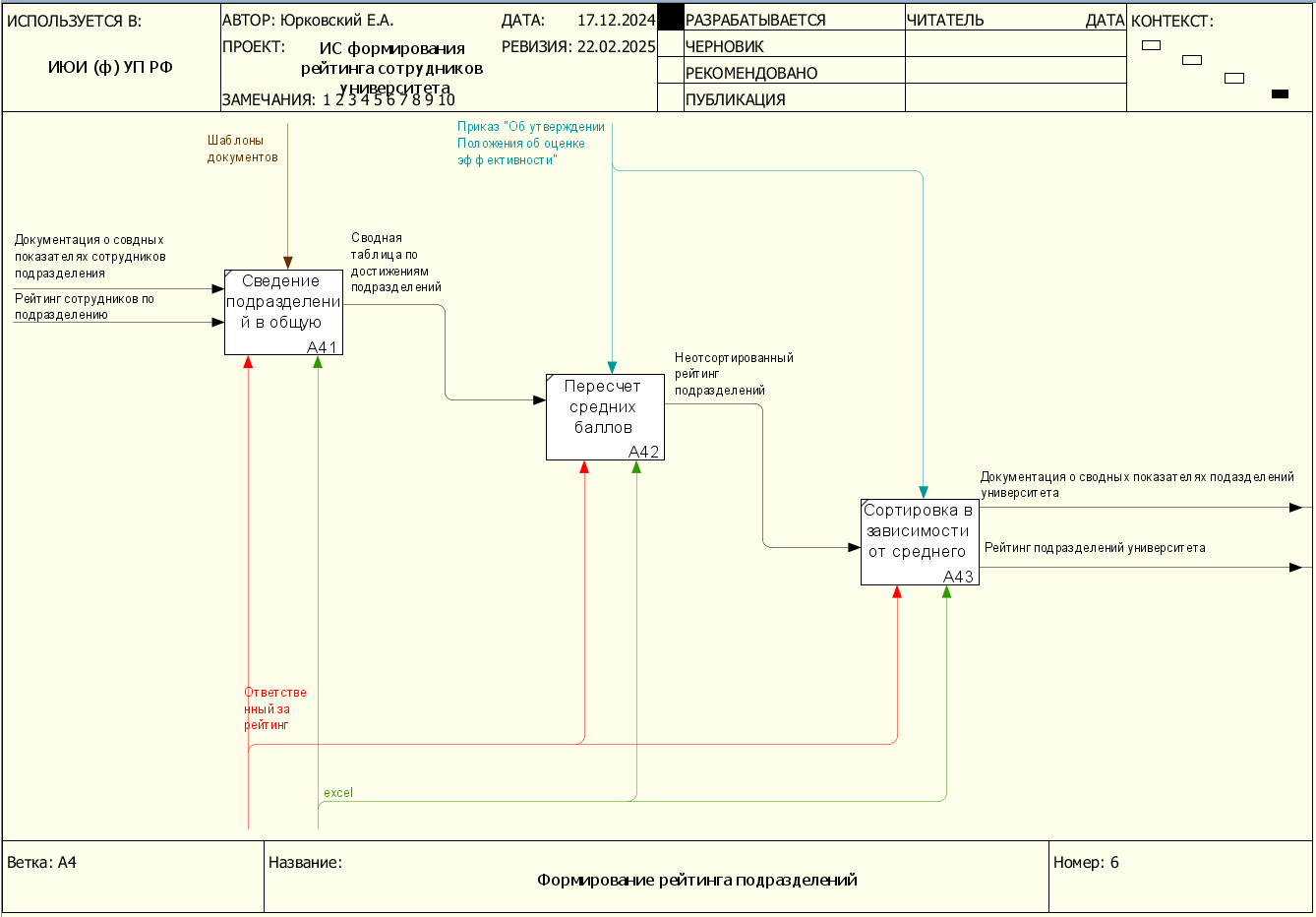


Рисунок 1.1.6 -Схема IDEF0 AS-IS. Уровень A4. Декомпозиция процесса формирования рейтинга подразделений

Исходя из смоделированного процесса можно выявить много неоптимальных и проблемных моментов, которые могут быть автоматизированы. Постоянный ручной расчет баллов создает большую нагрузку на ответственных за рейтинг, а необходимость постоянного документооборота влечет риск утери документов, что еще больше замедлит процесс.

Описанный процесс усложняет работу ответственных за составление рейтинга в период заполнения рейтинга, забирая на себя значительное количество времени, и приводит к ряду проблем с нехваткой времени для других, не менее важных, задач. Такой подход демонстрирует низкую эффективность процессов документооборота и расчета KPI, а также существенно усложняет контроль и анализ результатов. Сотрудники кадровых и учебно-методических подразделений вынуждены тратить свои силы на механическую работу, что отвлекает их от более важных задач.

## 1.2 Обзор существующих аналогов ИС «Эффективного контракта»

«Система эффективного контракта» — это общегосударственная практика, которая внедрена в большое количество ВУЗов страны, однако во многих учебных заведениях процесс составления итогового рейтинга все еще является ручным. В некоторой части университетов имеется ИС для автоматизации этого процесса и имеет различные функциональные возможности, которые довольно сильно отличают одну систему от другой. Большинство таких ИС представлены в качестве Веб-приложения, для удобства пользователей и составителей, ввиду отсутствия необходимости работы со сторонним приложением, а просто взаимодействия с вебсайтом в привычном для пользователя браузере. Однако большая часть таких сайтов скрыта от пользователя и доступ к ним может иметь только сотрудник университета, что усложняет анализ аналогов. Рассмотрим ИС эффективного контракта, которые представлены в различных университетах родного города заказчика – Иркутске.

На текущий момент времени подобные ИС уже давно внедрены и функционируют в нескольких вузах города Иркутска, а именно «ИРНИТУ» и «ИрГУПС».

ИС «ИрГУПС» находится в закрытом доступе и доступ для обычного пользователя к ней ограничен и доступен строго через личный кабинет сотрудника.

Рассмотрим информационную системы «Система эффективного контракта научно-педагогических работников ИРНИТУ», которая представлена в качестве веб-приложения:

Исходя из общения с разработчиками данной системы, она рассчитана, но большое число пользователей. Достижения заполняют различные структурные подразделения, которые относятся к этим достижениям. Преподаватель практически не может влиять на свои достижения, на которые не подал. Рейтинг практически не редактируется, вне сроков его формирования, внутренняя функциональность которой не выставляется в общий доступ.

Система имеет минималистичный интерфейс, фильтры по дате, а также строку поиска в шапке.

Основной частью страницы является таблица рейтинга. Рейтинг можно выбрать по подразделению или персональный. Регулируется удобными вкладками. Каждое поле таблицы предполагает сортировку в обе стороны. Стартовый интерфейс ИС «Система эффективного контракта научно-педагогических работников ИРНИТУ» представлена на рисунке 1.2.1.

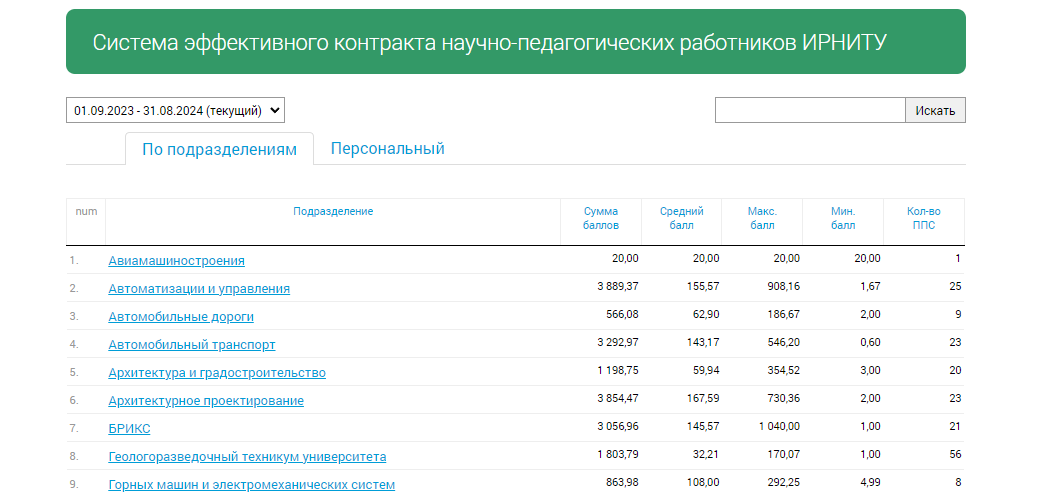


Рисунок 1.2.1 – Интерфейс ИС «Система эффективного контракта научно-педагогических работников ИРНИТУ», рейтинг подразделений

При переходе на подразделение выводится такая же таблица, но для преподавателей кафедры. Страница рейтинга преподавателей по подразделению представлена на рисунке 1.2.2.

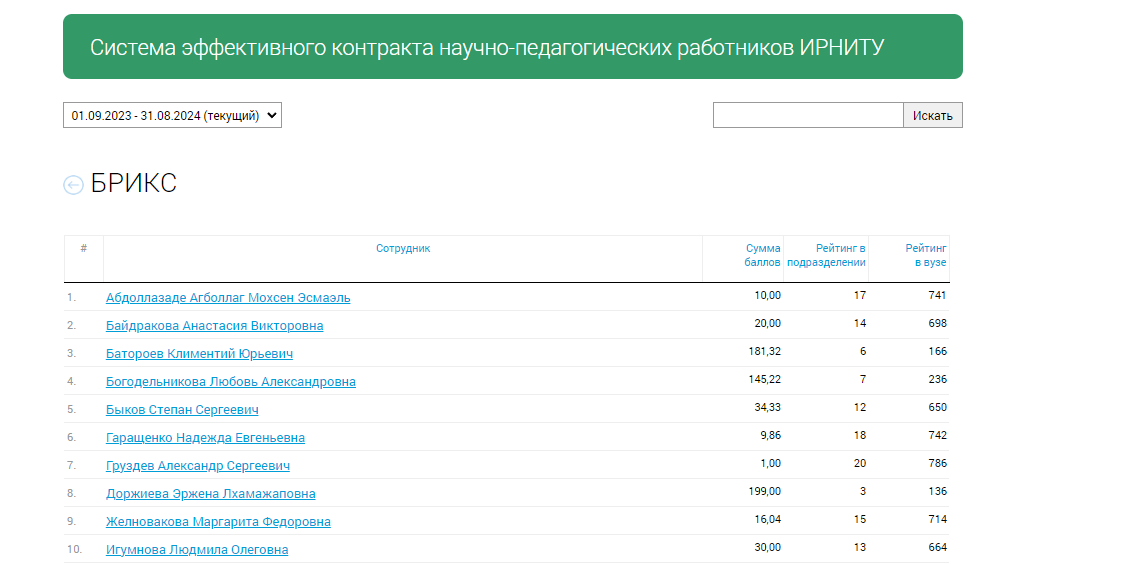


Рисунок 1.2.2 – Страница рейтинга преподавателей подразделения

При переходе на преподавателя выводится список всех критериев оценки, с выделением жирным тех критериев, по которым у преподавателя есть балл, а также указанием баллов в крайнем правом положении строчки критерия. Страница списка достижений преподавателя представлена на рисунке 1.2.3.

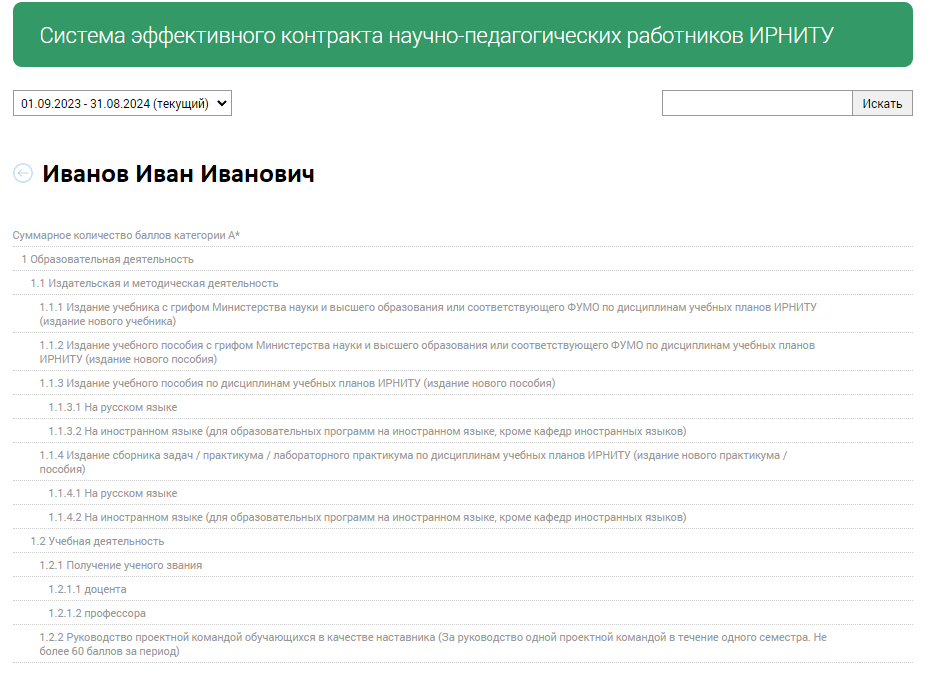


Рисунок 1.2.3 – Страница списка достижений преподавателя

В данной информационной системе есть также возможность посмотреть на конкретные достижения по показателю. На конкретном примере это реализовано не совсем удобно, а также, хоть и интуитивно понятно, но не для всех пользователей. Для того, чтобы увидеть, какие достижения есть у сотрудника, необходимо найти строчку с заполненным значением баллов, а затем задержаться курсором на ней на 2-3 секунды. Появится маленькое всплывающее окошко, на котором мелким шрифтом будет отображены конкретные достижения по данному показателю. Достижения будут идти друг за другом, а балл по каждому будет указан в скобочках. На рисунке 1.2.4 представлен пример просмотра достижения в ИС эффективного контракта ИРНИТУ

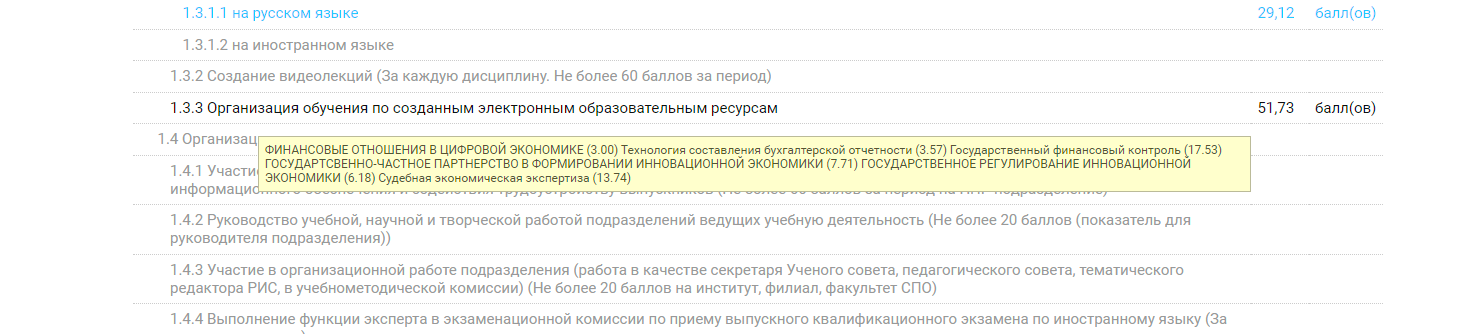


Рисунок 1.2.4 – Отображение достижений по показателю

Исходя из осмотра и оценки системы, были выявлены важные структурные моменты для будущего разрабатываемого приложения.

Было решено, что за идейную основу будет взят рассматриваемая информационная система эффективного контракта ИРНИТУ.

Важней частью стало принятие решения о повторении формата информационной системы в качестве веб-приложения. Данный формат удобен и доступен для сотрудников, а также прост в использовании, так как не нужно будет устанавливать специализированное ПО на каждый компьютер, а просто открыть веб-страницу в браузере.

Удобное отображение подразделений и сотрудников в таблицах позволит упросить процесс сортировки, а также приятен и понятен обычному пользователи. Таблицы позволяют эффективно выводить и структурировать всю информацию.

Внешний вид подкупает своей минималистичностью и функциональностью, поэтому было решено взять за основу основные моменты, например, такие как вкладки переключений рейтинга между сотрудниками и подразделениями.

Возможность перейти к списку сотрудников при нажатии на подразделение также является очень удобным, с точки зрения UX, шагом.

Удобная строка поиска и выбор даты, на который составляется рейтинг также было решено реализовать в ИС для заказчика.

Также были выявлены те недостатки, которые конкретная система не покрывает нужды заказчика.

Детальный список по достижениям в виде маленького всплывающего окошка, которое появляется при наведении на показатель и удержании курсора на протяжении 2-3 секунд является абсолютно неудобным решением. Достижения в этом маленьком окне расположены друг за другом и отделяются только их баллами, которые размещены в скобочках. Такой способ отображения является неэффективным и, вероятнее всего, неудобным для пользователя, потому необходимо переработать способ отображения достижений по показателям.

У стороннего наблюдателя за рейтингом нет возможности узнать, есть ли в данной ИС доступ в личный кабинет, либо в системе он отсутствует. Так как в ИЮИ (Ф) УП РФ сотрудники сами будут вносить свои достижения, и ответственными она будет только модерироваться, наличие прямого доступа в ЛК, т.е. место просмотра и внесения достижений, является критически важным.

Отсутствие автоматические генерируемой отчетности, гибкость в изменении расчетных формул и, самое важное, процесс заполнения такого рейтинга организован таким образов, что его заполняют те структурные подразделения, к которым относятся достижения, что в свою очередь не применимо для института маленького масштаба, коим является заказчик.

## 1.3 Постановка задачи

Разрабатываемая информационная система эффективного контракта предназначена для формирования рейтингов сотрудников и подразделений института. Ключевыми решаемыми проблемами будет авто расчет KPI и внутренний оборот данных, без вынесения их во вне в промежуточных этапах.

В настоящее время рутинная работа по расчету значительно усложняет работу сотрудников в период заполнения рейтинга, забирая на себя значительное количество времени, и приводит к ряду проблем с нехваткой времени для других, не менее важных, задач. Такой подход демонстрирует низкую эффективность процессов документооборота и расчета KPI, а также существенно усложняет контроль и анализ результатов. Сотрудники кадровых и учебно-методических подразделений вынуждены тратить свои силы на механическую работу, что отвлекает их от более важных задач. Данная система предназначена для упрощения и ускорения процесса формирования рейтинга согласно системе эффективного контракта.

**Целью** выпускной квалификационной работыявляется снижение нагрузки на ответственных за процесс формирования рейтинга сотрудников университета ИЮИ (ф) УП РФ в рамках «Системы эффективного контракта» за счет внедрения ИС, которая будет способна на автоматический перерасчёт KPI сотрудников университета, централизованное хранение данных о списках достижений сотрудников, формирование итоговых рейтингов по университету, по подразделениям, внутри подразделений, а также генерацию отчётных документов, согласно шаблонам, что повысит точность, сократит время и упростит деятельность в периоды формирования рейтинга.

**Задачи**, которые необходимо выполнить, для успешного достижения целипредставленной выпускной квалификационной работы:

* проанализировать текущий процесс формирования рейтинга в ИЮИ (ф) УП РФ;
* спроектировать новый бизнес-процесс «Формирование рейтинга университета в рамках системы эффективного контракта» с применением разрабатываемой ИС;
* cпроектировать систему – архитектуру, базу данных, серверную и клиентскую части, интерфейс;
* реализовать спроектированную ИС;
* провести интеграцию и развёртывание ИС на мощностях ИЮИ (ф) УП РФ;
* организовать и провести обучение сотрудников университета.

Клиентская часть системы в удобной форме выводит все представленные в базе данных сведения, которые отражают рейтинги сотрудников, подразделений и т.д. Благодаря удобному отображения в формате таблиц, данные легко сортируются по колонкам, а также фильтруются по дате и поисковой строке. Интерфейс построен интуитивно понятным способом, где нажатие на элемент таблицы, позволяет провалиться в него глубже, для уточнения необходимой информации.

Серверная часть системы обеспечивает взаимодействие пользователя через клиентскую часть и базы данных, в которой отражаются совершенные пользователем изменения, и которая предоставляет на клиентскую часть информацию, необходимую пользователю. Для этого используются API запросы. Также серверная часть будет ответственна за формирование уведомлений об удалении и редактировании достижений, которые будут направляться пользователям на электронную почту.

Информационная система разрабатывается как веб-приложение. Взаимодействие с системой осуществляется через API-запросы. Веб-клиент отправляет GET-запросы, POST-запросы, DELETE-запросы и UPDATE-Запросы к API для получения информации, внесения ее в базу данных, удаления или обновления. При взаимодействии с данными в базе данных через API система обрабатывает запросы, преобразует в необходимые форматы и сохраняет результат в базе данных.

Для хранения и доступа к актуальным данным о сотрудниках, показателях, подразделениях и т.д. система интегрирована с базой данных, которая хранит всю необходимую информацию в формате таблиц, которые отражают сущности, представленные в проекте. Система обеспечивает возможность быстрого поиска и извлечения информации, а также работе по внесению, изменению и удалению данных из нее.

Система также способна выводить отчеты, согласно шаблонным регламентированным документам, что в значительной степени упрощает работу по их ручному составлению. Форматами отчета может являться .xls, .xlsx и .pdf.

Результатом данной работы станет автоматизированная информационная система, которая будет способна вести все расчеты, связанные с процессом формирования рейтинга, централизовать процесс, при помощи хранения всех данных системы в одном месте, а также иметь возможность генерировать всю необходимую отчетность, в значительной степени упростит процесс формирования рейтинга сотрудников университета и значительно снизит нагрузку участвующих в этом процессе лиц, особенно в периоды активного составления рейтинговых списков. Кроме того, система устранит человеческий фактор, путем отсутствия необходимости документооборота между этапами процесса создания рейтинга и позволит с легкостью получать связанные с рейтингом данные за прошлые периоды, без необходимости поднимать архивы бумаг.

## 1.4 Требования к разрабатываемому обеспечению

### 1.4.1 Общие требования

**Требования к режимам работы:**

Подсистема должна поддерживать работу в двух основных режимах: штатном, администрирования.

**Штатный режим** представляет собой основной режим функционирования, при котором подсистема выполняет задачи работы с заполнением, корректировкой и просмотром рейтинга в полном объеме, позволяет генерировать отчеты и авторизовываться пользователям выполняя весь предоставленный функционал. В этом режиме система работает в оптимальных условиях, обеспечивая стабильность и высокую производительность.

**Режим администрирования** – это режим, предназначенный для поддержки и обслуживания подсистемы. В этом режиме осуществляется настройка различных параметров системы, например изменение критериев оценивания или корректировка формул расчета итога по показателю, проведение регламентных процедур обслуживания, таких как настройка резервного копирования данных и обеспечение безопасности программно-технической части, а также регистрацию новых пользователей и внесение данных о новых сотрудниках в базу данных. В общем весь режим можно привести к использованию панели администратора, для точечной работы с данными в базе данных.

**Требования по обеспечению доступности данных подсистемы интернет**

Должна быть обеспечена работа, как минимум, всего персонала университета в любом из режимов работы и ролей пользователя, а именно, примерно 100 едино моментных пользователей в режиме просмотра информации и, также, работа в режиме внесения и редактирования списков достижений для такого же числа пользователей. Также необходимо предоставить администратору возможность работать одновременно с работой системы. Доступ для открытого просмотра информации в сети Интернет должен быть строго ограничен авторизацией и не позволять стороннему человеку проникать в ИС посредством внутренних ссылок, которые, для неавторизованного пользователя будут перенаправлять на страницу авторизации. Любое взаимодействие с системой должно осуществляться СТРОГО авторизованным пользователем.

**Требования к численности и квалификации персонала системы**

Пользователи системы разделены на следующие группы:

* Сотрудник;
* Заведующий кафедры;
* Администратор;
* Ответственный за составление рейтинга.

**Сотрудник.** Пользователь системы, имеющий свою учетную запись, который вносит свои достижения, а также имеет полный доступ ко всем рейтинговым спискам университета. Список функциональных возможностей пользователя «Сотрудник»:

* Просмотр всех доступных в ИС рейтингов;
* Внесение своих достижений;
* Просмотр списка достижений каждого сотрудника;
* Просмотр детальной информации по каждому достижению.

**Заведующий кафедрой.** Пользователь системы, имеющий свою учетную запись с расширенным функционалом относительно пользователя с уровнем доступа **Сотрудник**, который, дополнительные возможности, к уже описанным в роли Сотрудник. Редактировать списки достижений сотрудников, относящихся к его подразделению путем изменения информации достижения либо его удаления. Также имеет возможность генерации отчетов по сотрудникам своего подразделения и генерации отчета по своему подразделению, сводную таблицу показателей сотрудников подразделения. Список функциональных возможностей пользователя «Заведующий кафедры»:

* Просмотр рейтинга;
* Внесение своих достижений;
* Корректировка достижений сотрудников, приписанных к подразделению;
* Удаление достижений сотрудников, относящихся к его подразделению;
* Генерация отчетных документов по каждому сотруднику, относящемуся к его подразделению;
* Генерация отчетного документа по подразделению.

**Ответственный за составление рейтинга.** Эта роль отводится тому, кто имеет наивысшие права в системе. Кроме всех вышеописанных функций ответственный проводит контроль за всеми внесенными достижениями. Кроме возможности корректировки и удаления абсолютно всех достижений, также имеет возможность генерации отёчности по каждому из рейтингов:

* Контроль за корректностью достижений;
* Удаление и редактирование всех достижений;
* Печать любой отчетной документации.

**Администратор.** Обеспечивает работоспособность системы, исправляет возникающие ошибки, вносит корректировки в критерии оценивания, при их обновлении, проводит работу по сопровождению системы, также отвечают за резервные копирования информации. Также важным является то, что только администратор может создать аккаунт для сотрудника, для обеспечения безопасности системы. Список функциональных возможностей администратора:

* Доступ к изменению критериев оценивания;
* Ведение резервного копирования информации системы;
* Сопровождение системы, исправление возникших ошибок;
* Регистрация и выдача данных новым пользователям.

**Требования к безопасности и надёжности**

Подсистема должна обладать надежностью, обеспечивающей работу пользователей в круглосуточном режиме в периоды активной работы с расчетом KPI, но, также, иметь доступ и вне расчетных периодов. Восстановление работоспособности при сбоях должно производиться в кратчайшие сроки, особенно в периоды заполнения рейтинга.

С целью обеспечения безопасного функционирования в подсистеме должны быть предусмотрены:

* Контроль целостности данных на уровне СУБД;
* Регулярные резервные копирования данных;
* Защита от вредоносных данных;
* Защита от ввода некорректной информации (SQL-инъекций);
* Отсутствие доступа к ИС для неавторизованных пользователей.

Сохранение работоспособности программного обеспечения при некорректных действиях пользователя (ввод неверных по размеру и/или составу данных, нештатное завершение действий и т.п.).

Система должна исключать возможность перехода в систему по внутренним ссылка, обходя авторизацию, так как данные должны быть доступны только авторизованным пользователям.

Система должна обеспечивать высокий уровень **надежности** за счет использования механизмов регулярного резервного копирования данных. Одно обязательное резервное копирования после окончания периода формирования рейтинга, для точного сохранения данных, а также регулярное проведение резервного копирования (2 раза в сутки) в периоды формирования рейтинга. Это позволяет восстановить актуальные данные в случае сбоев и минимизировать риск их потери. Должна быть предусмотрена функция автоматического сохранения всех критически важных данных при любых сбоях системы, чтобы гарантировать их сохранность и возможность восстановления без значительных потерь.

### 1.4.2 Функциональные требования

**Задачи, которые выполняет ИС можно разделить по ролям пользователей:**

**Сотрудник**

**1. Внесение достижения (является одной из основных задач ИС):**

* + Сотрудник выбирает показатель из списка всех показателей [2];
  + Сотрудник выбирает подкатегорию показателя из списка [2];
  + Сотрудник прикрепляет подтверждающие документы в зависимости от типа подтверждающего документа / условия начисления баллов [2]. Система поддерживает загрузку файлов различных форматов (PDF, DOC, JPEG, PNG, DOCX, XLS, XLSX, TXT).

**2. Просмотр списка достижений сотрудника (Является одной из основных задач ИС):**

* + Сотрудник может просматривать свой собственный список достижений;
  + Любой сотрудник может просматривать списки достижений любого сотрудника;
  + Список достижений отображается в порядке кода показателя (подкатегории показателя);
  + При необходимости сотрудник или администратор может открыть детальную информацию о конкретном достижении, включая скачивание прикрепленного документа;
  + Можно воспользоваться строкой поиска для нахождения нужного достижения;
  + Можно получить список достижений за нужный период формирования рейтинга.

**3. Просмотр рейтинга сотрудников всего университета (Является одной из основных задач ИС):**

* + Рейтинг формируется на основе суммарного балла KPI сотрудников;
  + Рейтинг доступен для просмотра всем сотрудникам, но с ограниченными правами (без возможности редактирования и удаления для пользователей с ролью «Сотрудник»);
  + Пользователь может выбрать отсортировать рейтинг по KPI, по фамилиям сотрудников, по месту в рейтинге;
  + Пользователь может открыть рейтинг за другой период составления рейтинга.

**4. Просмотр рейтинга сотрудников по подразделению (Является одной из основных задач ИС):**

* + Рейтинг формируется на основе суммарного балла KPI сотрудников;
  + Рейтинг доступен для просмотра всем сотрудникам, но с ограниченными правами (без возможности редактирования и удаления для пользователей с ролью «Сотрудник»);
  + Пользователь может выбрать отсортировать рейтинг по KPI, по фамилиям сотрудников, по месту в рейтинге;
  + Пользователь может открыть рейтинг за другой период составления рейтинга.

**5. Просмотр рейтинга подразделений всего университета (Является одной из основных задач ИС):**

* + Рейтинг формируется на основе среднего балла KPI по сотрудникам подразделения;
  + Рейтинг доступен для просмотра всем сотрудникам, но с ограниченными правами (без возможности редактирования и удаления для пользователей с ролью «Сотрудник»);
  + Пользователь может выбрать отсортировать рейтинг по KPI, по названиям подразделений, по месту в рейтинге;
  + Пользователь может открыть рейтинг за другой период составления рейтинга.

**Заведующий кафедрой**

«Заведующий кафедрой» является ролью с расширенными правами и включает всю функциональность роли «Сотрудник», а также дополнительные функциональные возможности:

**1. Редактирование достижений (Является одной из основных задач ИС):**

* + Заведующий кафедрой может редактировать достижения сотрудников своей кафедры;
  + Доступны операции обновления, удаления и подтверждения достижений;
  + При завершении редактирования достижения сотруднику направляется уведомление на электронную почту.

**2. Генерация отчета по сотруднику (Является одной из основных задач ИС):**

* + Отчет содержит полную информацию о достижениях сотрудника [4], Отчет может быть экспортирован в формате Excel или PDF;
  + заведующий кафедрой может инициировать генерацию отчета по сотрудникам, которые относятся к его кафедре.

**3. Генерация отчета по подразделению (Является одной из основных задач ИС):**

* + Отчет содержит информацию по итоговым баллам по каждому показателю и подпункту показателя для всех сотрудников подразделения [3], Отчет может быть экспортирован в формате Excel или PDF;
  + Заведующий кафедрой может инициировать генерацию отчета только по своему подразделению.

**Ответственный за формирование рейтинга**

«Ответственный за формирование рейтинга» является ролью с расширенными правами и включает всю функциональность роли «Сотрудник», а также дополнительные функциональные возможности:

**1. Редактирование всех достижений (Является одной из основных задач ИС):**

* + Ответственный за формирование рейтинга может редактировать и удалять все достижения;
  + Доступны операции обновления, удаления всех достижений;
  + При завершении редактирования достижения сотруднику направляется уведомление на электронную почту.

**2. Генерация отчета по сотруднику (Является одной из основных задач ИС):**

* + Отчет содержит полную информацию о достижениях сотрудника [4], Отчет может быть экспортирован в формате Excel или PDF;
  + Ответственный за формирование рейтинга может инициировать генерацию отчета по всем сотрудникам.

**3. Генерация отчета по подразделению (Является одной из основных задач ИС):**

* + Отчет содержит информацию по итоговым баллам по каждому показателю и подпункту показателя для всех сотрудников подразделения [3], Отчет может быть экспортирован в формате Excel или PDF;
  + Ответственный за формирование рейтинга может инициировать генерацию отчета по всем подразделениям.

**Администратор**

Администратор является ролью, которая не участвует в процессе формирования самого рейтинга, но является определяющим звеном всей ИС, потому что именно этот человек наполняет внутреннюю БД данными о новых сотрудниках, регистрирует им аккаунты в ИС, обновляет баллы за показатели и сами показатели и т.д.

* + - * Редактирование, удаление и внесение информации о сотрудниках, подразделениях и показателях в БД;
      * Регистрация новых пользователей в системе.

**Требования к протоколам и форматам передаваемых данных**

**Протоколы передачи данных:**

Для передачи информации между клиентом и сервером используются протоколы HTTP и HTTPS, SMTP. HTTP обеспечивает базовый обмен файлами и обработку запросов, тогда как HTTPS гарантирует защиту данных, используя шифрование, что делает его стандартом безопасного взаимодействия, а также обеспечивая формирование и отправку email сообщений.

**Форматы передаваемых данных:**

1. JSON — применяется для передачи структурированных данных в API-запросах. Содержит параметры настройки, идентификаторы задач и сообщения об ошибках;
2. PDF —итоговые документы формируются в формате PDF [3, 4]. Для долгосрочного хранения и корректного отображения итоговых документов в формате PDF используется стандарт PDF/A, поддерживающий настройку графических параметров и совместимость с различными устройствами;
3. Excel — итоговые документы формируются в формате EXCEL [3, 4].

**Методы взаимодействия**

Взаимодействие клиента и сервера организовано через REST API, обеспечивающий загрузку данных, получение результатов, настройку параметров вывода и управление функционалом системы.

**Требования к внутренним форматам данных системы**

Входной информацией ИС будут является данные, которые вносятся пользователем для заполнения своего достижения, а именно базовые типы данных (строки, числа, булевые) собранные в JSON формат для передачи в базу данных, а также разные расширения файлов для подтверждающей информации, которая выражена в файле. Например, для загрузки подтверждающего повышение квалификации сертификата можно загрузить его в формате PDF, PNG, JPG, JPEG и формате ZIP, если подтверждение требует пакет документов. Необходимо обеспечить поддержку, также, для офисных файлов (DOC, DOCX, XLS, XLSX, TXT). Хранение таких файлов необходимо обеспечить на сервере в отдельной директории.

Для хранения структурированных по таблицам данных использовать СУБД PostgreSQL, потому что ПО обеспечивает весь необходимый функционал, а также является отечественным продуктом, что важно для университета, как государственной структуры.

Выходной информацией ИС будут являться данные таблиц из базы данных, для отображения в интерфейсе, а также заполненный документ эффективного контракта преподавателя в форматах Excel и PDF.

**Схемы данных (метаданные), используемые в подсистеме**

**Необходимо хранить:**

**Данные о сотрудниках:**

* + ФИО;
  + Кафедра;
  + Должность.

**Данные о кафедрах:**

* + Название.

**Данные о достижениях:**

* + Наименование показателя;
  + Единица измерения [2];
  + Наименование;
  + Условие начисление баллов [2];
  + Дата;
  + Количество баллов.

Данные о группах достижений

* + Наименование;
  + Описание.

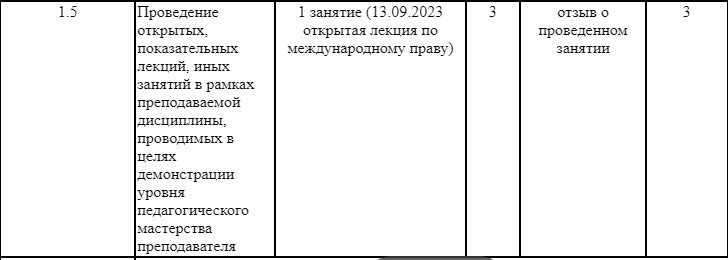


Рисунок 1.4.2.1 Пример достижения из приложения к приказу, на основе которого проектируется система.

* Данные о пользователя системы:
  + Логин;
  + Пароль;
  + Роль.

**Типовые запросы на получение данных** избазы данных**:**

* Получение списка рейтинга:
* По подразделению;
* По университету.
* Получение всех доступных достижений (В рамках добавления достижения);
* Получение стоимости достижения/ий;
* Получение списка достижений конкретного сотрудника;
* Получение списка каферд;
* Получение данных пользователя.

## Заключение по разделу

В ходе предпроектного исследования был проведен детальный анализ текущего процесса формирования рейтинга сотрудников ИЮИ (ф) УП РФ в рамках «Системы эффективного контракта». Было выявлено, что процесс в значительной степени ручной, что приводит к высокой трудоемкости, повторному вводу данных и риску ошибок. Основные проблемы включают в себя необходимость многократного переноса данных между различными таблицами и документами, а также отсутствие автоматизации расчетов KPI.

Был проведен обзор существующих аналогов информационных систем, таких как ИС «Система эффективного контракта научно-педагогических работников ИРНИТУ», что позволило выделить ключевые функциональные возможности и недостатки, которые необходимо учесть при разработке новой системы. На основе анализа были сформулированы требования к разрабатываемой системе, включая общие, функциональные и требования к безопасности.

Итогом предпроектного исследования стало понимание необходимости создания автоматизированной системы, которая позволит сократить время на формирование рейтинга, минимизировать человеческий фактор и повысить точность расчетов. Были определены основные задачи, которые необходимо решить для успешной реализации проекта, включая анализ текущего процесса, проектирование нового бизнес-процесса, разработку архитектуры системы и ее интеграцию в существующую инфраструктуру университета.

# 2 Проектирование ИС

## 2.1 Процесс формирования рейтинга с применением проектируемой ИС

Основная задача проектирования и разработки ИС для формирования рейтинг – это минимизировать работу человека в автоматизируемой области, сократить время на исполнение необходимых задач, а также повысить точность, путем исключения человеческого фактора. Также важной задачей, применимо к текущей предметной области, является централизовать документооборот, что в данной системе реализуется при помощи отмены большинства документов, ограничиваясь лишь итоговыми заполненными отчетами, которые будут заполняться данными, хранящимися в ИС. Как раз таки заменой всем документам между этапами процесса будет является обработка данных внутри системой самой машиной. Для оценки преимуществ была сформирована схема в идентичной описанию формирования рейтинга в текущее время, нотации – IDEF0. Общая схема практически идентична, в нее была добавлена ИС в качестве механизма. В схеме декомпозиции общего блока присутствуют значительные изменения в использовании механизмов. В блок А1 теперь приходит еще и ИС, а вот все остальные блоки теперь исполняются только системой, все же остальные механизмы, в число которых входят «Сотрудник», «Заведующий кафедрой», «Ответственный за составление рейтинга», а также инструмент учета всей информации теперь становятся не нужны для успешного завершения процессов. Также повышается точность, так как все расчеты проводить машина, которой не свойственно ошибаться, при заранее установленном, корректном алгоритме пересчета. Структура блоков же декомпозиции остается неизменной, просто устраняя исполнителей и необходимость использования шаблонов для заполнения. В итоговом варианте 3 из 4 процессов перехватывает на себя ИС, что является 75% заменой человека машиной в реализации данного процесса. Значительным улучшением процесса, являются авто формирующиеся email письма об изменении или удалении достижения сотрудника, что убирает необходимость лично идти и сообщать преподавателю, либо вручную создавать электронное сообщение. Ниже, на рисунках 2.1.1 и 2.1.2, представлены общая диаграмма процесса формирования рейтинга с применением проектируемой ИС в нотации IDEF0 и ее декомпозиция на подпроцессы.

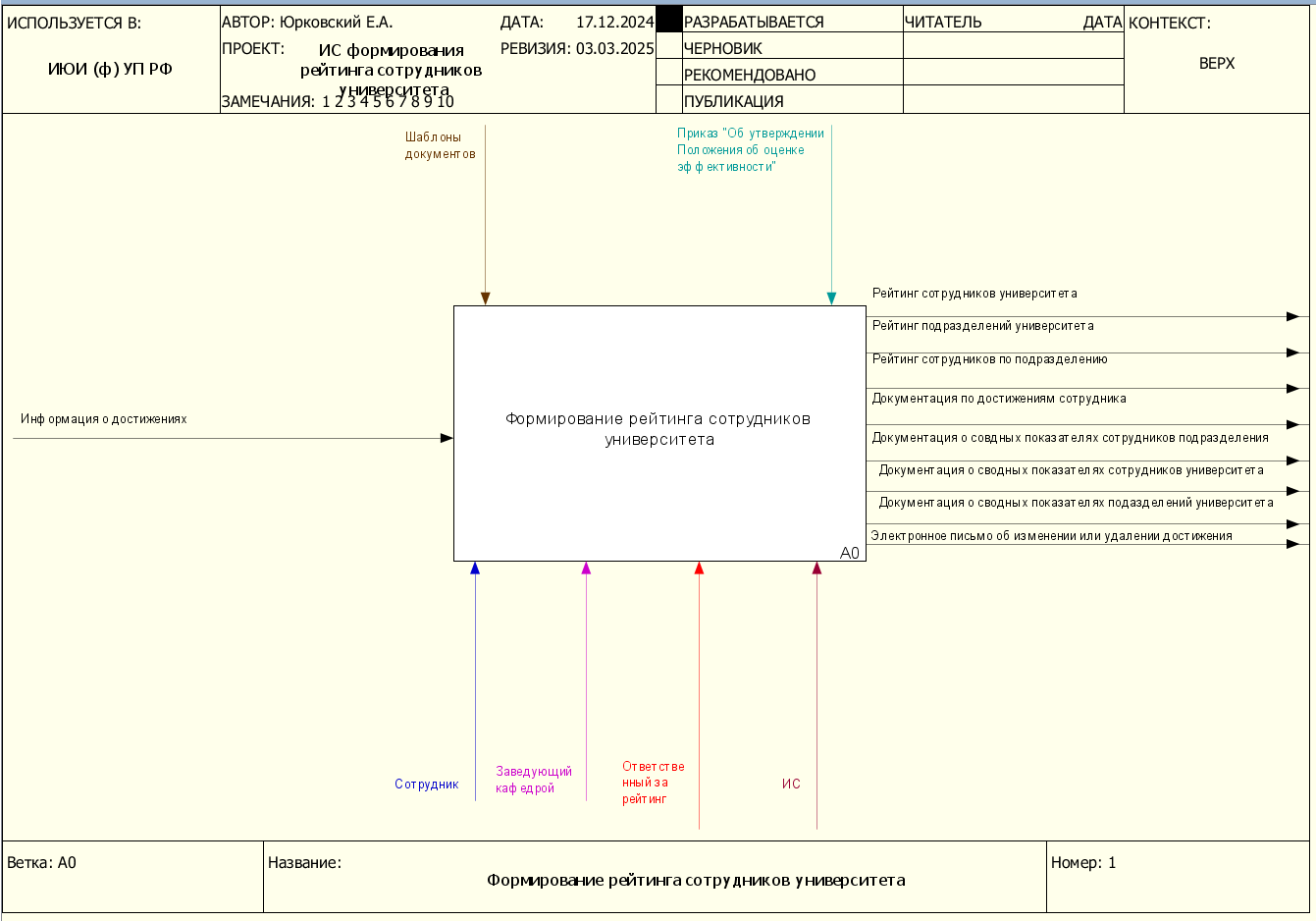


Рисунок 2.1.1 -Схема IDEF0 TO-BE. Уровень A-0. Формирование рейтинга сотрудников университета с применением проектируемой ИС

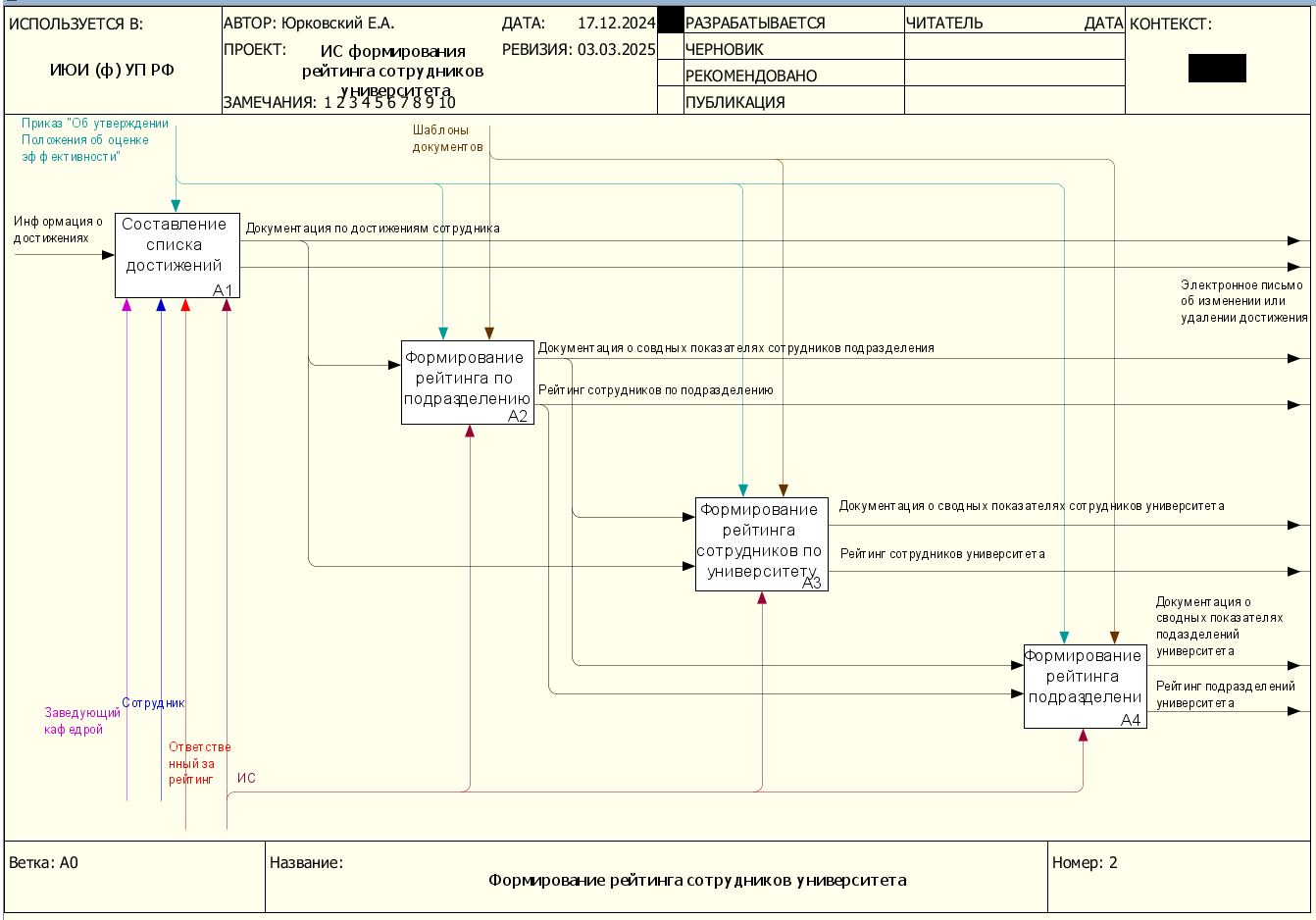


Рисунок 2.1.2 -Схема IDEF0 TO-BE. Уровень A0. Формирование рейтинга сотрудников университета с применением проектируемой ИС

Для того, чтобы заполнить достижения в проектируемой ИС, сотрудник должен внести по каждому показателю, который соответствует имеющемуся достижению, информацию, система автоматически подставит все баллы и рассчитает итог, что исключает ручной подсчет. После внесения всех имеющихся достижений ИС рассчитает общий балл (получившийся KPI), сотруднику необходимо лишь добавить к каждому из внесенных достижений подтверждающие документы / условия начисления достижений [2], для каждого подпункта показателя. После формирования своего списка достижений список автоматически уйдет на проверку «Ответственному за формирование рейтинга» и «Заведующему кафедрой» его подразделения. Ниже, на рисунке 2.1.3 представлена схема декомпозиции процесса составление списка достижений сотрудника.

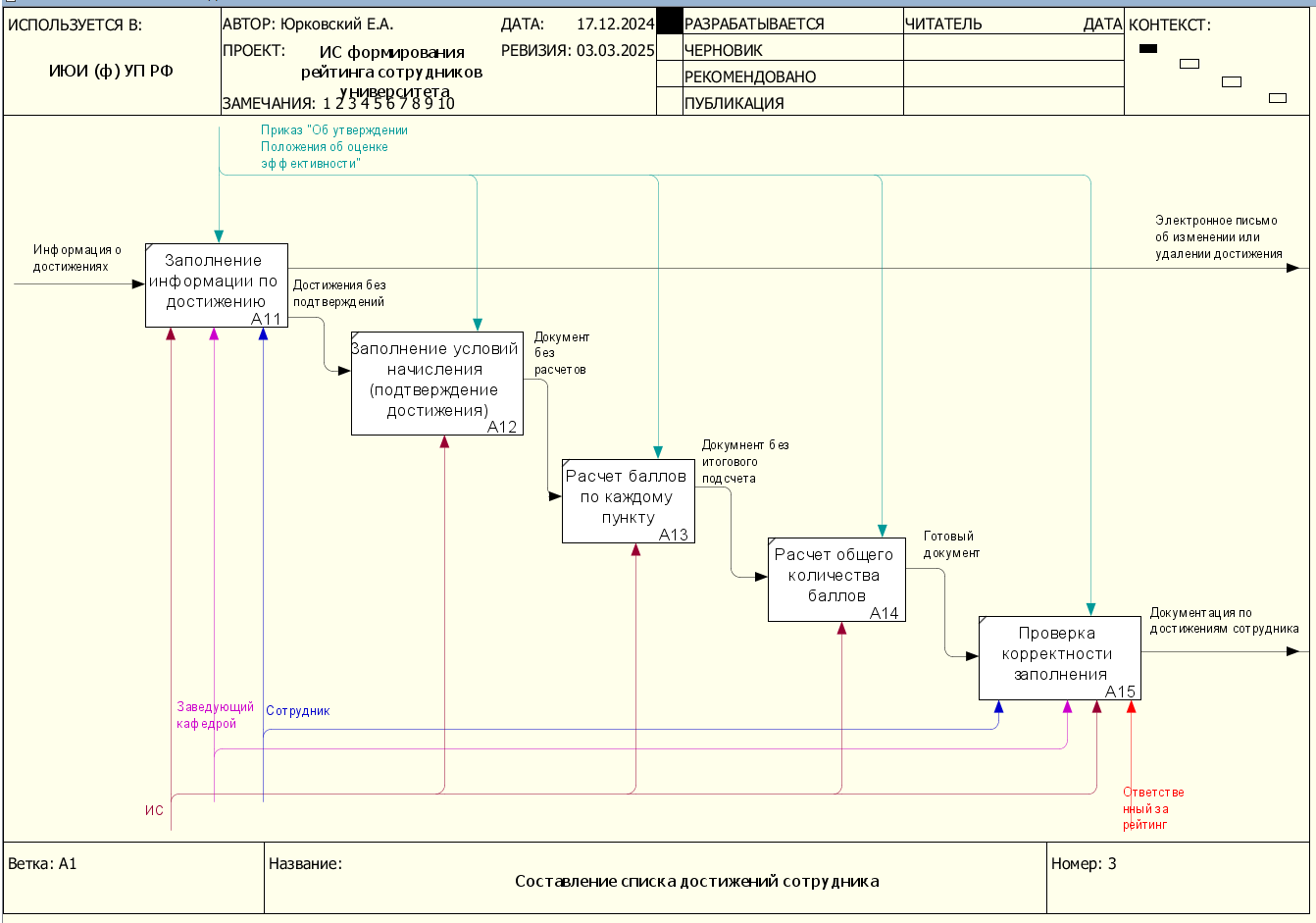


Рисунок 2.1.3 -Схема IDEF0 TO-BE. Уровень A1. Декомпозиция процесса составление списка достижений сотрудника

Все дальнейшие процессы структурно на уровне декомпозиции остаются неизменными, важным и значительным изменением является отсутствие в них, как механизмов всех элементов, кроме появившейся ИС. Исключение EXCEL и остальных механизмов убирает ручную работу из этих блоков полностью их автоматизируя. Ниже представлены рисунки 2.1.4, 2.1.5, 2.1.6, которые описывают процесс формирования рейтингов.

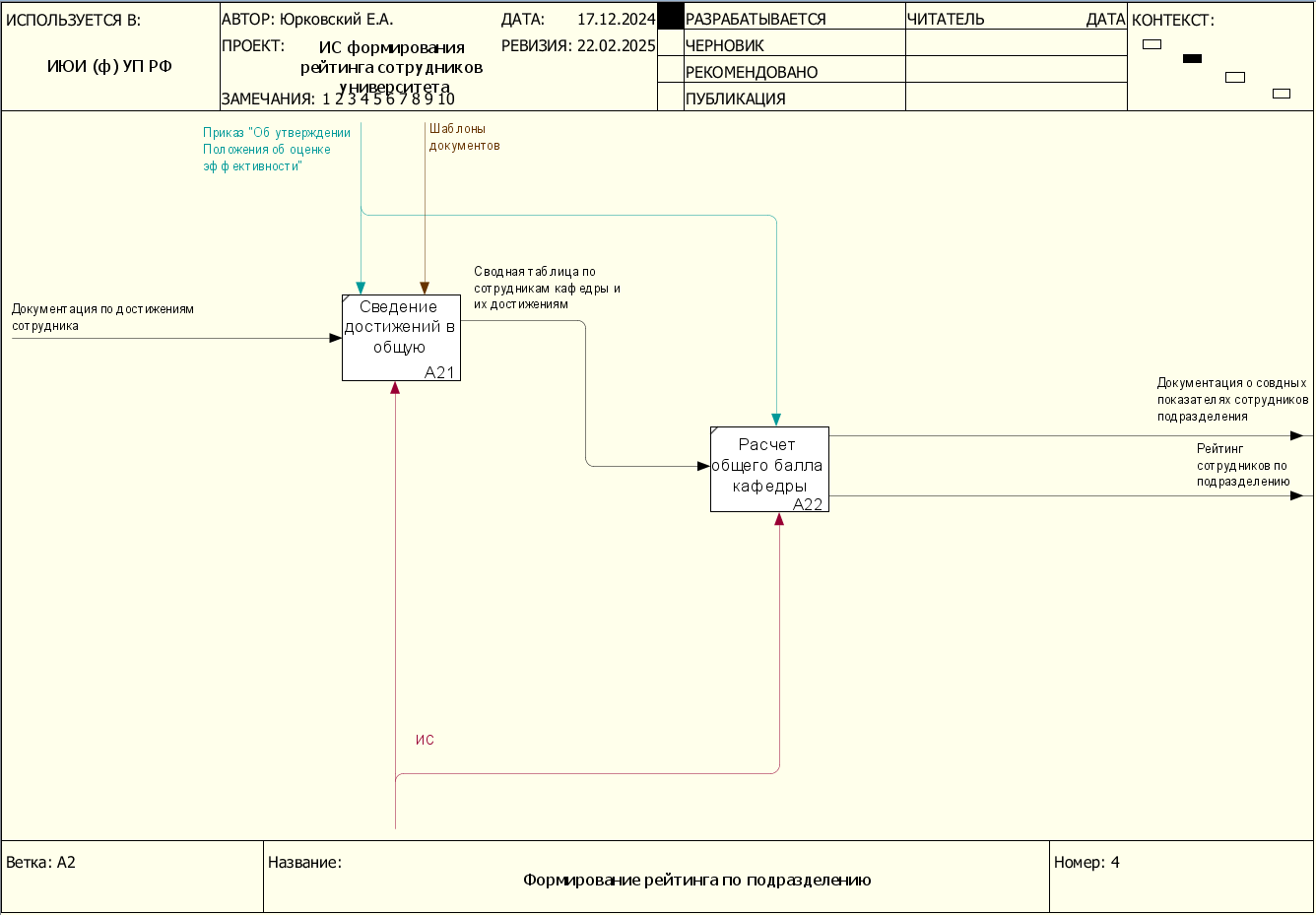


Рисунок 2.1.4 -Схема IDEF0 TO-BE. Уровень A2. Декомпозиция процесса формирования рейтинга по подразделению

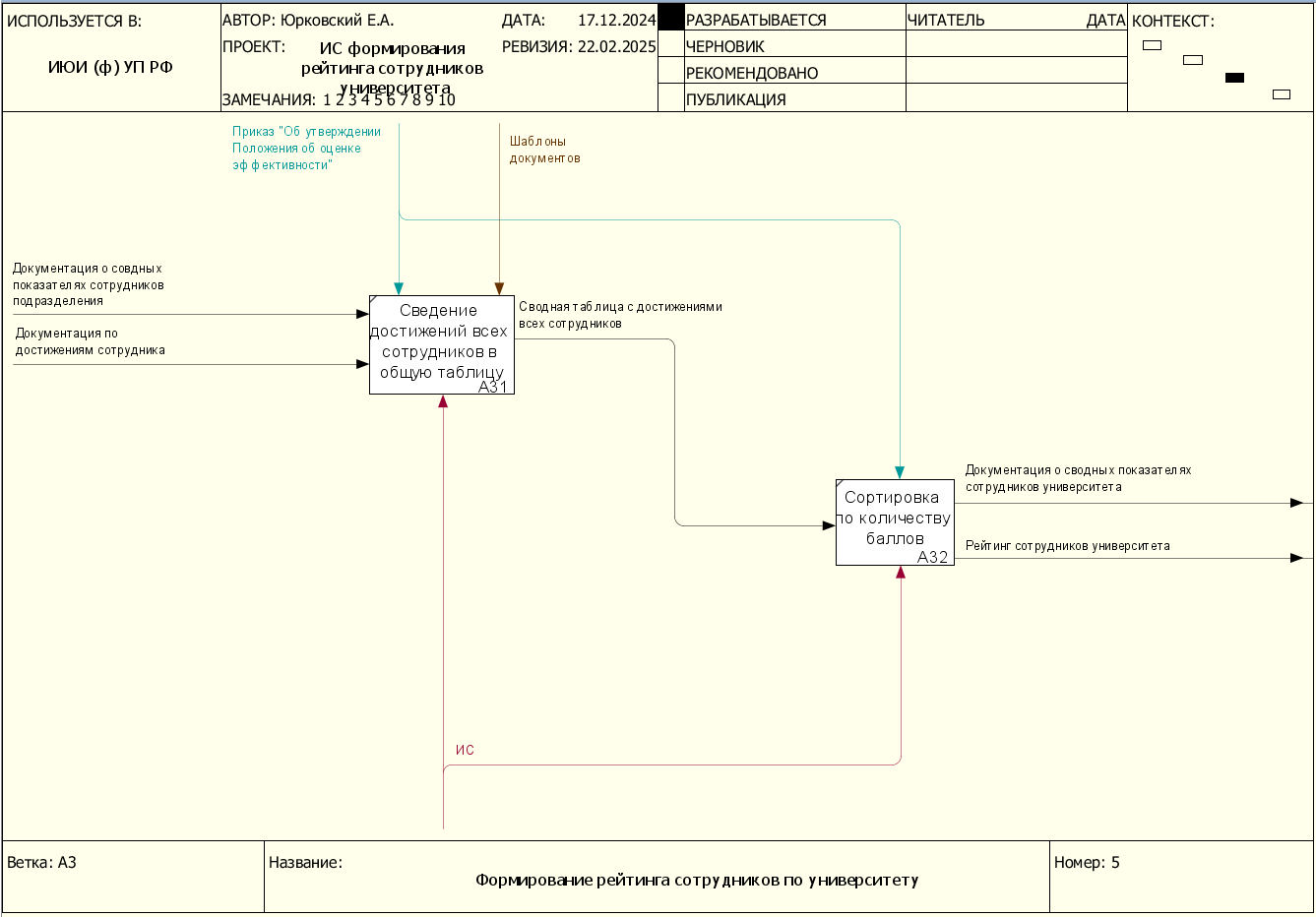


Рисунок 2.1.5 -Схема IDEF0 TO-BE. Уровень A3. Декомпозиция процесса формирование рейтинга сотрудников по университету

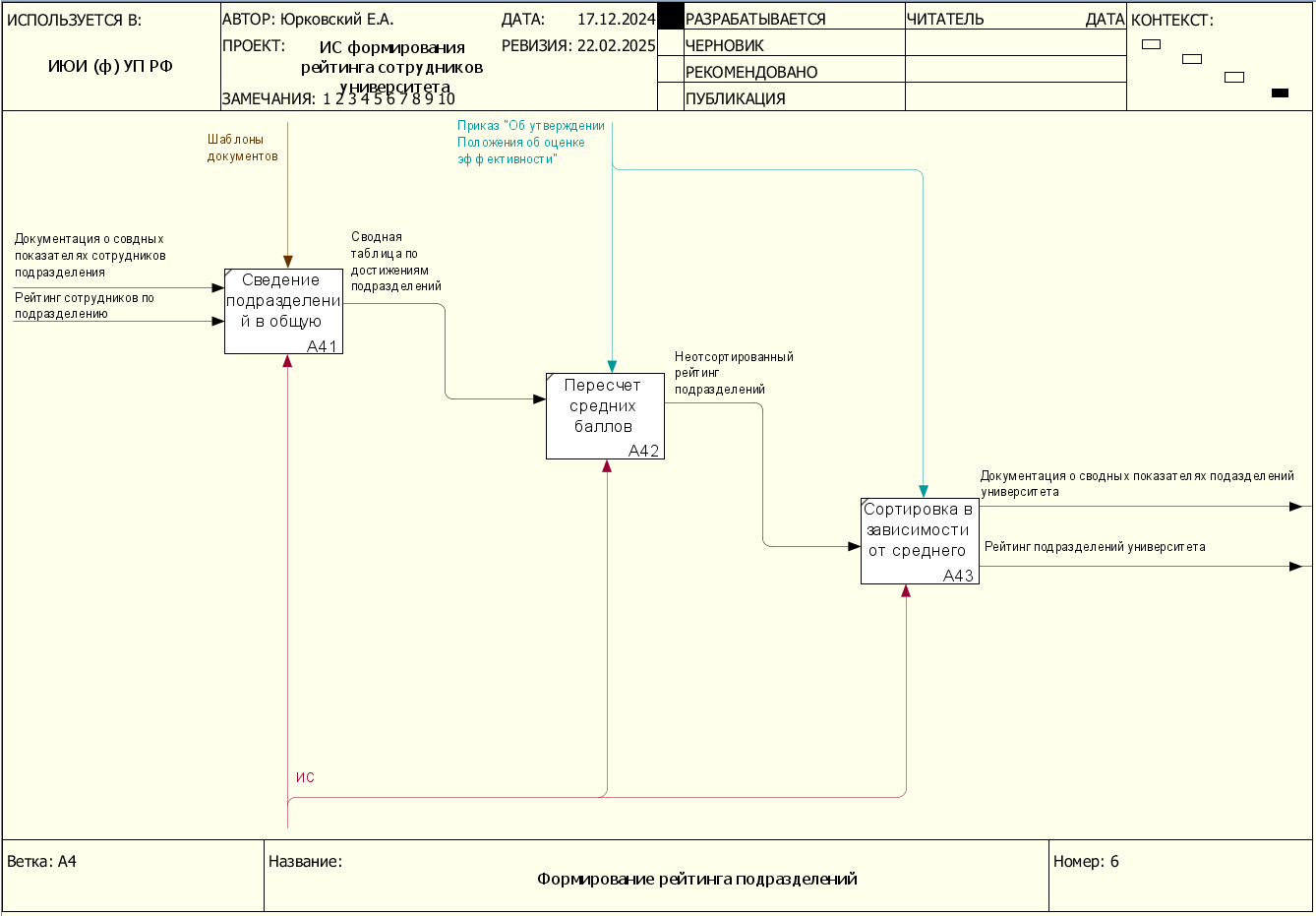


Рисунок 2.1.6 -Схема IDEF0 TO-BE. Уровень A4. Декомпозиция процесса формирования рейтинга подразделений

Итогом смоделированного бизнес-процесса будет являться полная автоматизация 3 из 4 основных блоков процесса формирования рейтинга, что означает, что 75% действий этого процесса теперь будет полностью подзадачной частью ИС вместо нескольких людей. Также полностью исключается работа с электронными таблицами, которые теперь будут использоваться только в качестве формата выгрузки из системы данный в рамках выходной документации, следовательно можно полностью исключить их из процесса. Столь значительный выигрыш в использовании человеческих ресурсов позволяет открыть возможности для исполнения других, более значимых дел.

## 2.2 Схема потоков данных ИС

Основной проблемой текущего процесса формирования рейтинга сотрудников университета является постоянный документооборот между лицами, участвующими в составлении. Необходимость переносить документы от человека к человеку значительно увеличивает время для того, чтобы завершить этап формирования. В разрабатываемой системе, в качестве замены промежуточным документам будет выступать оборот данных внутри самой ИС. Так, например, пользователи вносят данные о своих достижениях в систему, которая автоматически формирует список, рассчитывает баллы по показателям, общий балл сотрудника и подразделения, и выстраивает их в порядке, соответствующем рейтингу. Также информация из контура ИС может быть вынесена в качестве отчетов, необходимых для руководящих лиц, что будет являться результатом прошлого документооборота без всех промежуточных бумаг. В качестве новой роли процесса, администратор будет вносить информацию о пользователях в ИС, а также менять необходимые показатели, добавлять или убирать сотрудников, менять или редактировать названия подразделений и т.д. Все данные будут храниться в таблицах базы данных, которые будут символизировать собой определенные сущности, а также промежуточная информация будет содержаться во временных хранилищах на стороне клиента, для динамического формирования полноценных объектов. На рисунке 2.2.1 представлена схема подков данных на уровне А0 в нотации DFD:

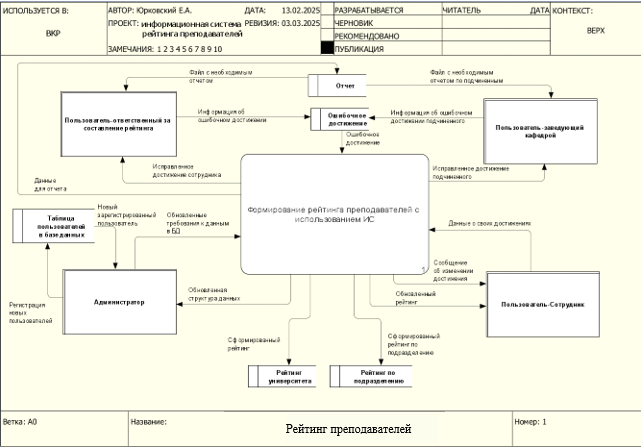
 

Рисунок 2.2.1 -Схема DFD потоков данных уровень A0

В рамках декомпозиции блока А0 представлены показаны основные процессы и данные, которые будут между ними переходить, а также элементы, в которых данные будут храниться в долгосрочном или коротко срочном формате. На рисунке 2.2.2 представлена схема подков данных декомпозиции уровня А0 в нотации DFD:

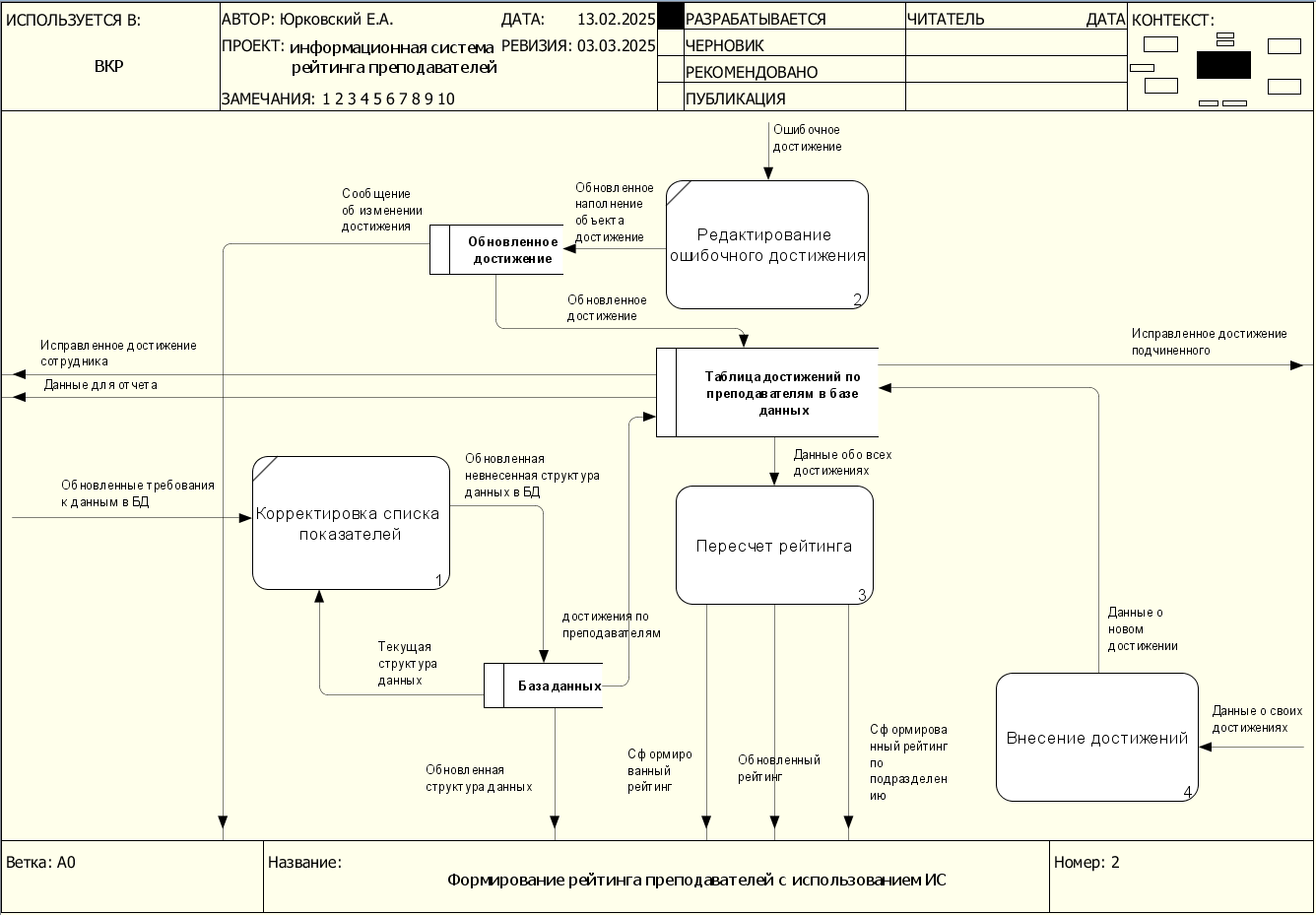


Рисунок 2.2.2 -Схема DFD потоков данных. Декомпозиция блока A0

На уровне А3 происходит процесс «Пересчета рейтинга», который затрагивает один из основных автоматизированных процессов, который нивелирует документооборот. На рисунке 2.2.3 представлена схема подков данных декомпозиции уровня А3 в нотации DFD:

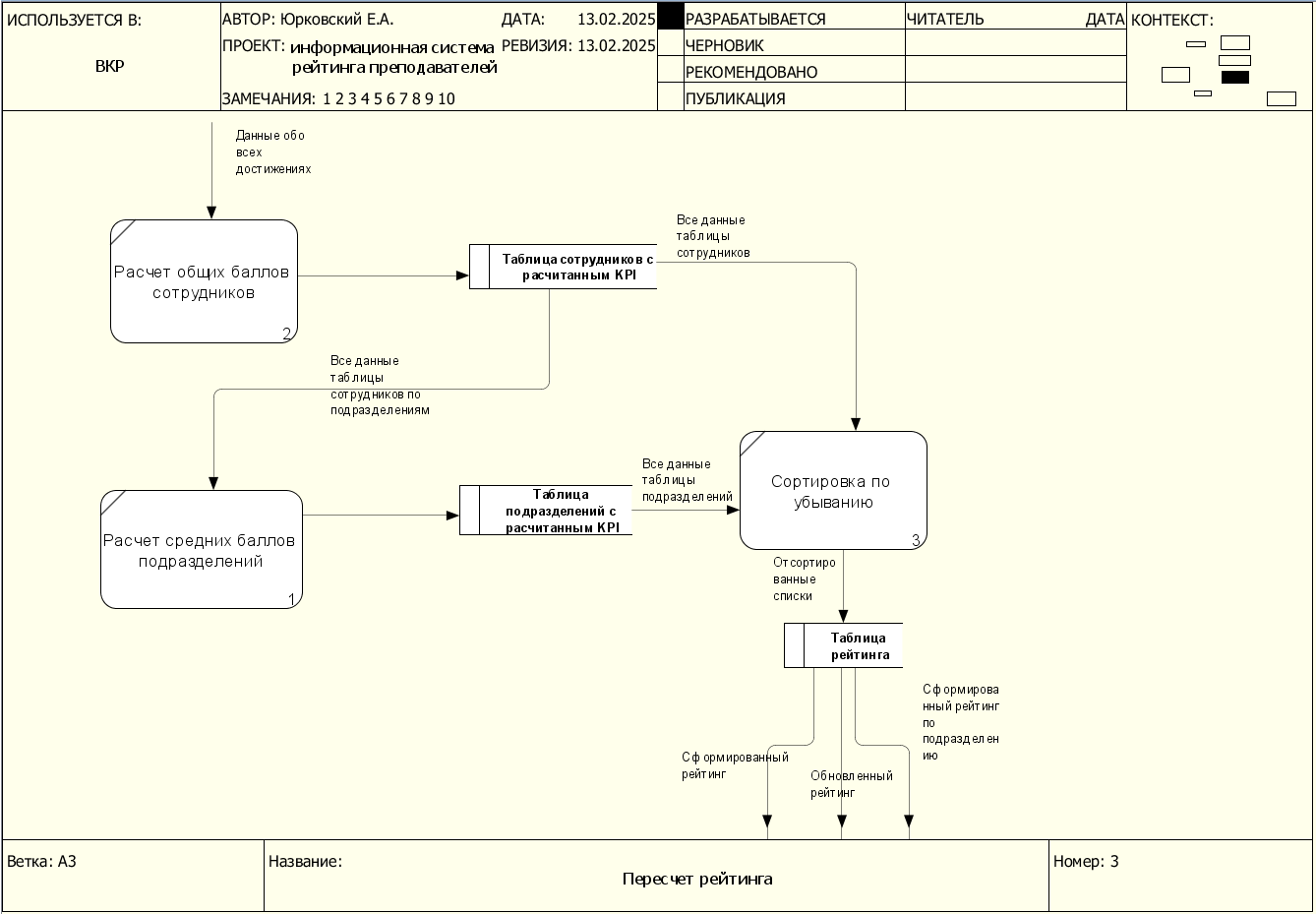


Рисунок 2.2.3 -Схема DFD потоков данных. Декомпозиция блока A3

Дальше, в блоке A4 показан процесс, в котором часто используются временные хранилища на стороне клиента для формирования полного объекта данных и отправки его в постоянное хранилище, в таблицу базы данных с достижениями сотрудников. На рисунке 2.2.4 показана декомпозиция блока А4 «Внесение достижения»:

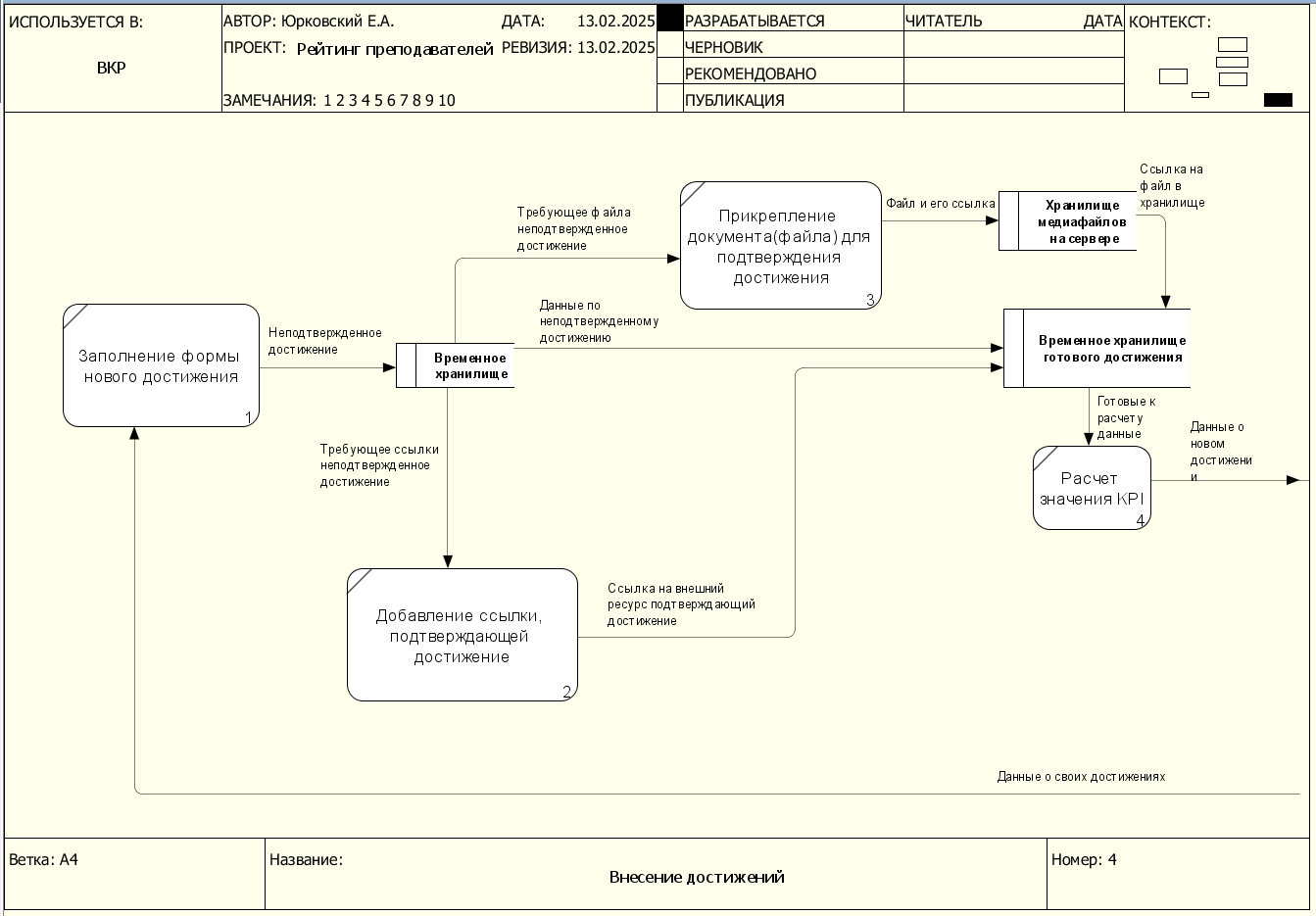


Рисунок 2.2.4 -Схема DFD потоков данных. Декомпозиция блока A4

## 2.3 Задача унификации показателей

Так как упрощение работы ответственных и сокращение времени на формирование ими рейтинга, это ключевая задача ИС необходимо предоставить им такую сущность достижения сотрудника, в которой будет внесена вся необходимая по достижению информация, которую останется только проверить и отредактировать или удалить, в случае некорректности данного достижения. Следовательно, для проектирования данной информационной системы эффективного контракта для ИЮИ (ф) УП РФ является необходимым создание модели данных, которые будут представлять собой сущности, которые будут отражать собой достижения, полученные конкретными сотрудниками. Недостаточно просто указать сотрудника, показатель и названия, так как в документах [2] требуется также указать условия начисления достижения, которых, по текущим вводным документам, можно выделить несколько:

Документ (или файл). Отражает достижение, например рецензия, отзыв, диплом об участии в конференции, учебный план, отчет, удостоверении о повышении квалификации, программа мероприятия и т.д. Самым важным из этой категории можно выделить тот факт, что любое из перечисленных условий можно прикрепить файлом

Ссылка на публикацию. Данных показателей немного и все эти показатели объединяет то, что согласно документам, их можно подтвердить ссылкой на источник размещения публикации, учебника, рецензии и т.д.

Исходя из этого можно выделить два реквизита у объекта, который будет характеризовать собой сущность, которая будет отвечать за накопление достижений конкретных преподавателей, а именно – реквизит подтверждающего документа и реквизит подтверждающей ссылки. Исходя из этого имеется 100% возможность полностью покрыть собой абсолютно все показатели, описанные в документе, определяющем эти самые показатели, для формирования рейтинга.

Данной информации по любому достижению будет достаточно для проверки ответственным и корректировки или удаления в случае несоответствия. К тому же ответственному нет необходимости самому уведомлять сотрудника, достижение которого было изменено или удалено, он получить на электронную почту об этом сообщение, и будет обязан принять меры, если хочет, чтобы достижение учитывалось.

## 2.4 Схема данных ИС

### 2.4.1. Инфологическая модель

После решения о виде самой важной сущности можно приступать к определению всех остальных сущностей и всех их реквизитов и как они связаны между собой. Для того, чтобы наглядно отразить все сущности, которые будут представлены в информационной системе эффективного контракта университета будет необходимо построить модель данных, наиболее наглядной из которых, для инфологического уровня, считаются ER-диаграммы.

Для предметной области «Рейтинг сотрудников университета» имеет смысл выделить следующие сущности ER-модели:

* Сотрудник;
* Пользователь;
* Подразделение;
* Достижения сотрудника;
* Достижение.

Каждой сущности есть основные атрибуты, которые описывают характеристики сущности, которые имеют значения в рамках описываемой предметной области и будущего разрабатываемого ПО. В текущем случае сущности имеют следующие атрибуты:

* Сотрудник:
  + Подразделение;
  + ФИО.
* Пользователь:
  + Логин;
  + Пароль;
  + Роль;
  + Сотрудник.
* Подразделение:
  + Название.
* Достижение:
  + Наименование;
  + Условие подтверждения;
  + Родитель
  + Стоимость единицы измерения;
  + Единица измерения;
  + Номер;
* Достижения сотрудника:
  + Итоговый балл;
  + Полное наименование;
  + Подтверждающий документ;
  + Подтверждающая ссылка;
  + Итоговое значение единицы измерения;
  + Дата.
* Группа достижений:
  + Наименование;
  + Описание;

Для наглядного представления взаимосвязей между выделенными сущностями и их атрибутами разработана концептуальная модель базы данных. На схеме, представленной на рисунке 2.4.1.1, отражены основные объекты предметной области, их атрибуты и связи между ними.

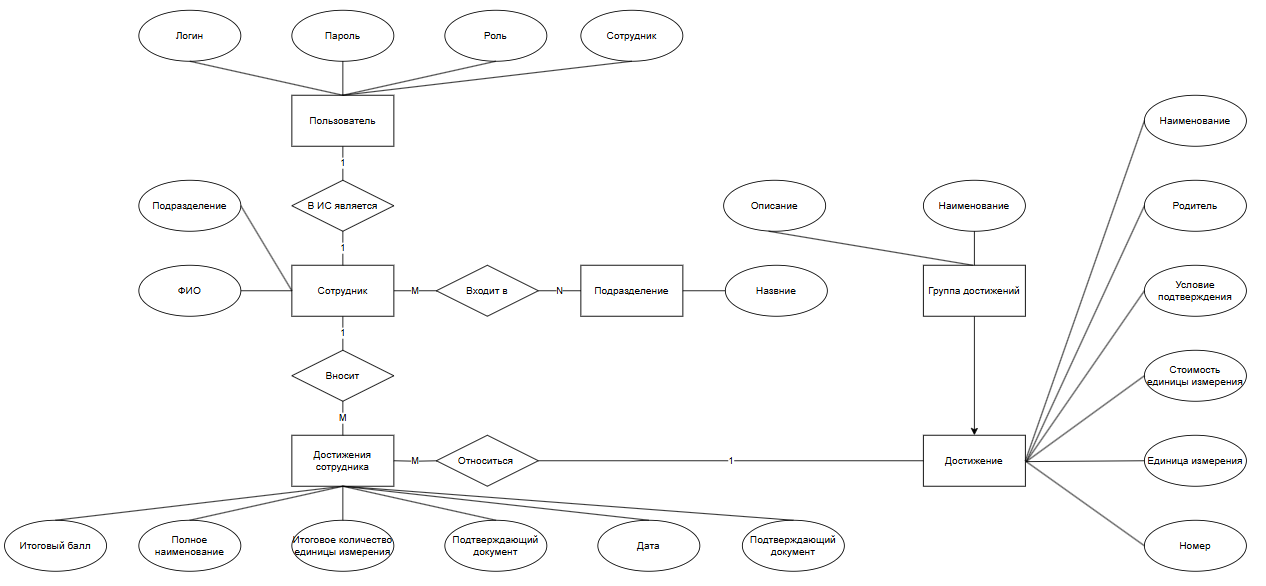


Рисунок 2.4.1.1 – Инфологическая модель

### 2.4.2. Физическая модель

На основе концептуальной модели была разработана логическая модель базы данных, которая представлена на рисунке 2.4.2.1 В процессе проектирования были применены принципы нормализации для устранения избыточности и обеспечения целостности данных.

Для хранения информации о достижениях сотрудников используется таблица Employee\_Achievment, связывающая сотрудников с их достижениями, которая является объектов, которые унифицирует все показатели, для более удобного и простого хранения и использование данных в дальнейшем.

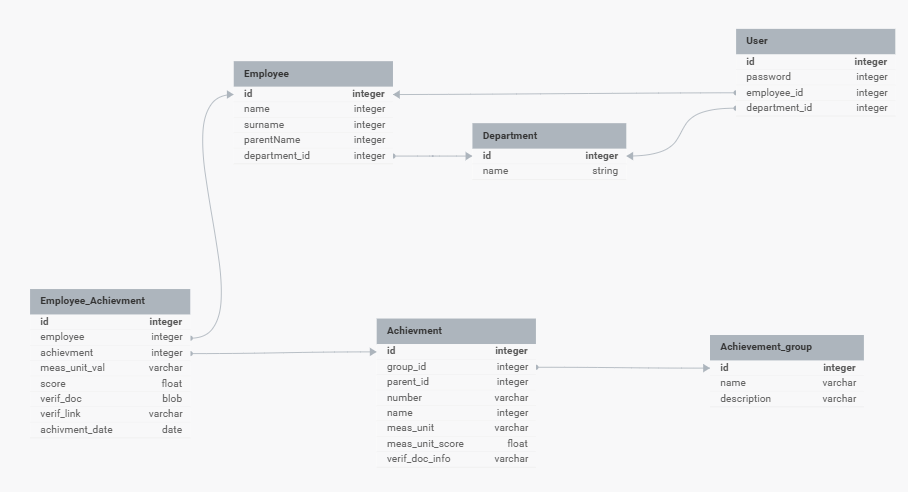


Рисунок 2.4.2.1 – Физическая модель

### 2.4.3. Модель БД в разрезе PostgreSQL модель

Физическая модель данных была создана на основе логической модели с учетом особенностей PostgreSQL. На этом этапе произошла конкретизация типов данных и реализация механизмов, обеспечивающих целостность и эффективность работы базы данных.

Одним из ключевых изменений стало уточнение типов данных в соответствии с возможностями PostgreSQL. Например, для идентификаторов сущностей (id) использован тип integer, который автоматически создает последовательность чисел и упрощает управление первичными ключами. Для текстовых данных применены varchar с ограничением длины или text, если размер строки заранее не известен. Числовые данные, такие как баллы за достижения, хранятся в типе double, а даты достижений — в date.

Также в физической модели были явно определены связи между таблицами с использованием FOREIGN KEY, что гарантирует ссылочную целостность данных. Это означает, что, например, запись о сотруднике в таблице Employee\_Achievment не может существовать без соответствующего сотрудника в Employee и достижения в Achievment.

Для повышения надежности данных в некоторых столбцах были добавлены NOT NULL-ограничения, запрещающие хранение пустых значений там, где они недопустимы. Например, у пользователей всегда должны быть логин и пароль, а у достижений — название и баллы. В ряде случаев были введены UNIQUE-ограничения, например, для логинов пользователей, чтобы избежать дублирования учетных записей. На рисунке 2.4.3.1 представлена модель базы данных физическая, сформированная в PostgreSQL, на основании созданных таблиц базы данных:

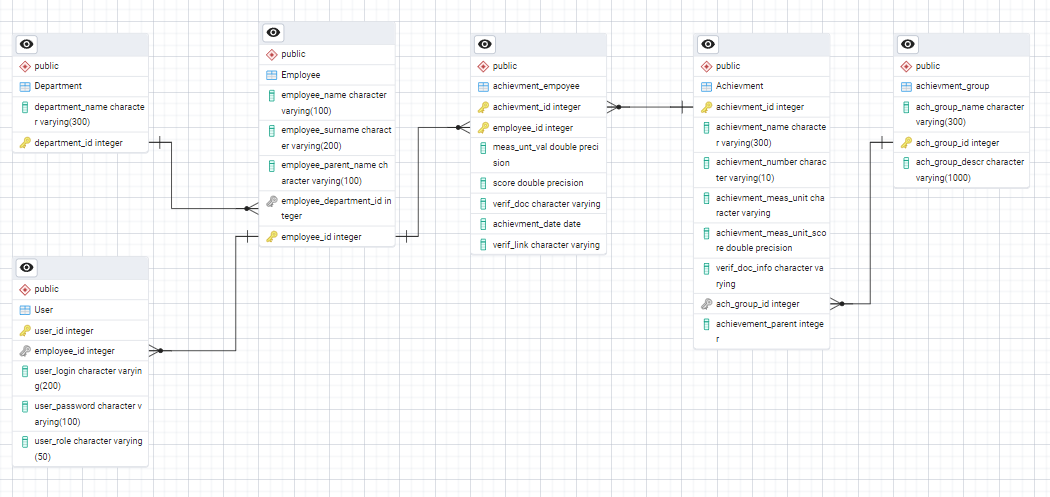


Рисунок 2.4.3.1 – Модель БД в PostgreSQL

## 2.5 Описание макетов интерфейса.

Наиболее важной, для пользователей, частью программы, исключая технический функционал программы, является удобство ее использования и то, как она выглядит. Удачный дизайн и UX позволяет превзойти такой же продукт с повышенным функционалом, но непродуманным интерфейсом.

Страница, которая приветствует пользователя при переходе на сайт (Запуске ИС) является страница авторизации. Так как система предназначена для внутреннего пользования, весь контент доступен только авторизованным пользователям. При попытке проникновения в систему неавторизованному пользователю по внутренним ссылкам, ИС всегда будет отправлять пользователя на страницу авторизации. Система не предполагает самостоятельной регистрации пользователем, поскольку аккаунты для них буду выдаваться тех. Специалистом (Администратором). Ниже на рисунке 2.5.1 представлен макет интерфейса «Страница авторизации»:

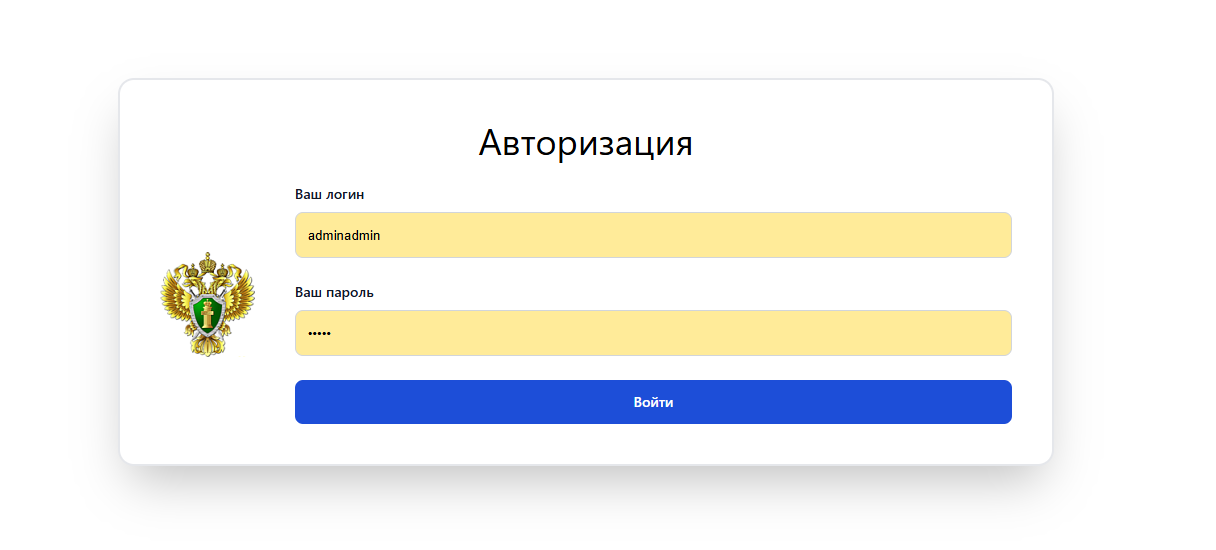


Рисунок 2.5.1 – Макет интерфейса страницы авторизации

После успешной авторизации пользователь автоматически будет переброшен на вкладку с общим рейтингом по подразделениям университета. Поскольку рейтинг должен составляться честно, то он выстраивает места согласно среднему баллу. Вкладка рейтинга «По подразделениям» явно указывает пользователю о каком рейтинге в таблице в данный момент идет речь. Кнопка в шапке является кнопкой перехода в личный кабинет. Таблица имеет возможность сортировки при нажатии на заголовки шапки, о текущем столбце и порядке сортировки будет сообщать стрелочка у названия колонки. Также ИС поддерживает фильтрацию данным по поиску и по периоду формирования рейтинга, за это отвечают поля фильтрации под шапкой. На рисунке 2.5.2 представлен макет интерфейса рейтинга по подразделениям университета:

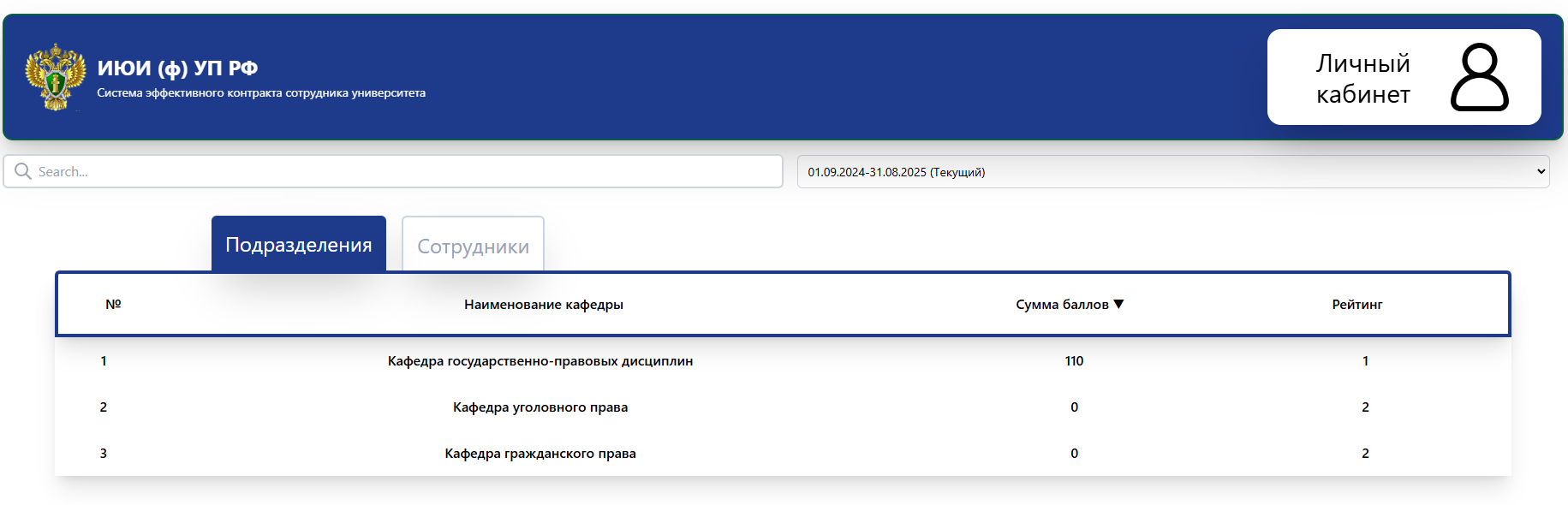


Рисунок 2.5.2 – Макет интерфейса страницы рейтинга по подразделениям (кафедрам)

У основного окна рейтинга есть вторая вкладка - «Персональный». В ней представлен рейтинг сотрудников университета. Иерархия в нем строится уже по общему баллу сотрудника. На рисунке 2.5.3 представлен макет интерфейса персонального рейтинга сотрудников университета:

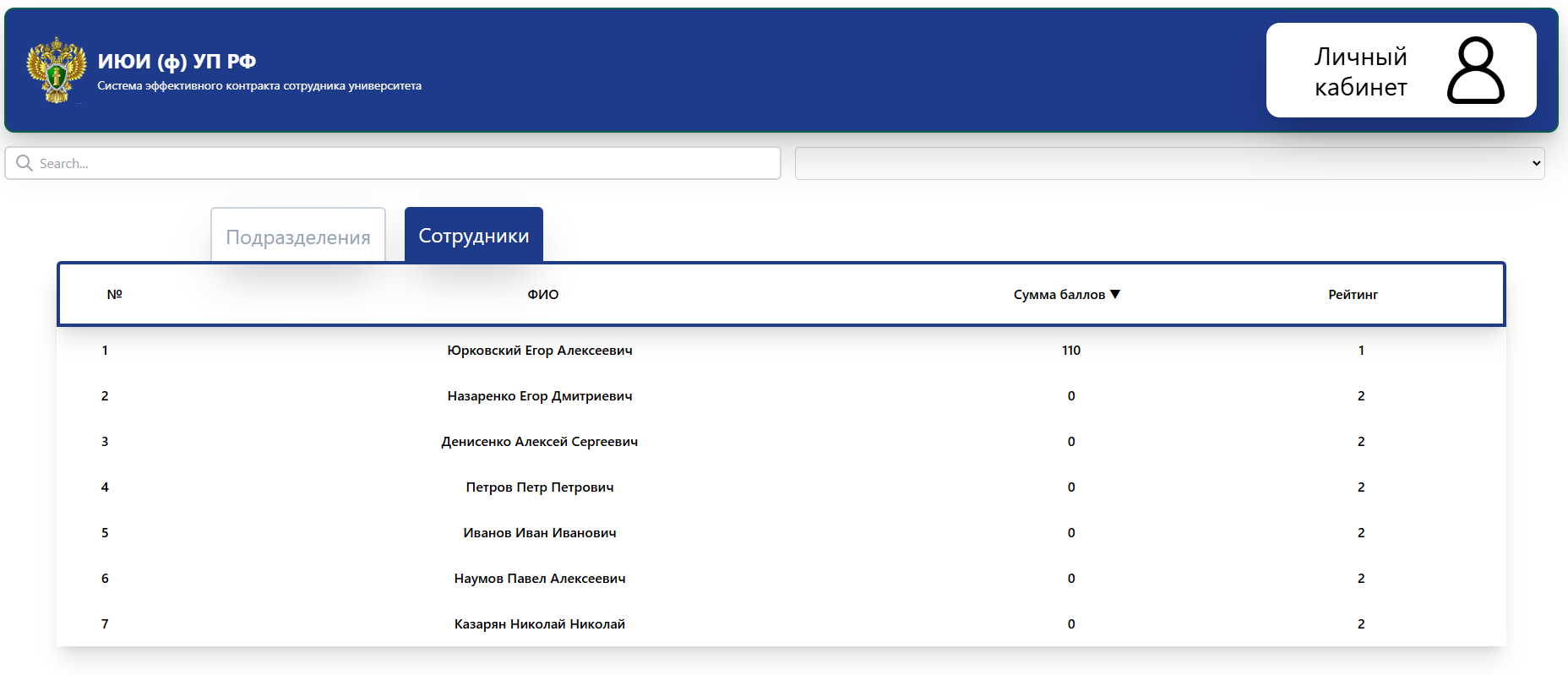


Рисунок 2.5.3 – Макет интерфейса страницы персонального рейтинга сотрудников университета

При нажатии на подразделение (кафедру) происходит переход Для авторизованного пользователя с правами доступа «Заведующий кафедрой» на странице сотрудников его кафедры есть кнопка распечатать отчетность, при нажатии на которую будет скачан сгенерированный файл отчета в формате Excel. На рисунке 2.5.4 представлен макет интерфейса страницы рейтинга сотрудников университета по подразделению:

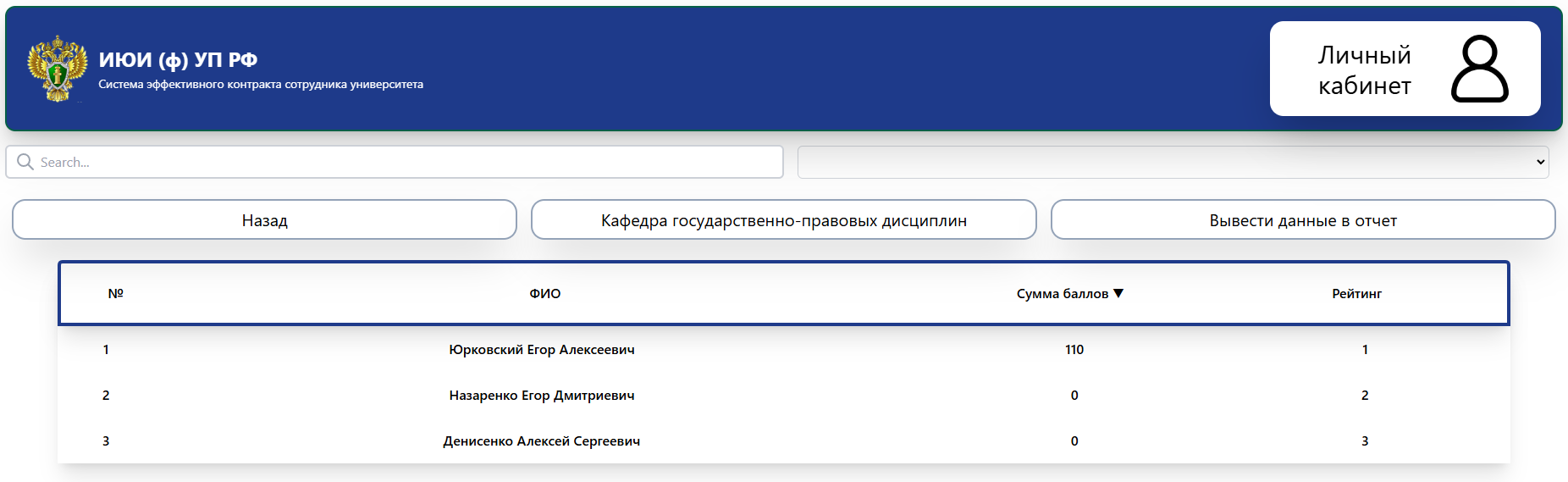


Рисунок 2.5.4 – Макет интерфейса страницы рейтинга сотрудников университета по подразделению

При нажатии на сотрудника происходит переход на страницу достижений сотрудника. На ней представлен список всех возможных показателей. Показатели, по которым сотрудник заполнял достижения, будут иметь проставленный бал, который является суммой баллов по достижениям данного показателя. Для авторизованного пользователя с правами доступа «Заведующий кафедрой» на странице достижений сотрудника его кафедры есть кнопка отчетность, при нажатии на которую будет скачан сгенерированный файл отчета в формате Excel. На рисунке 2.5.5 представлен макет интерфейса страницы достижений сотрудника:

Рисунок 2.5.5 – Макет интерфейса страницы достижений сотрудника

При нажатии на показатель открывается окно достижений по данному показателю. В нем представлен список всех достижений, которые к нему относятся, и информация по ним. На рисунке 2.5.6 представлен макет интерфейса для страницы «Достижение по показателю»:

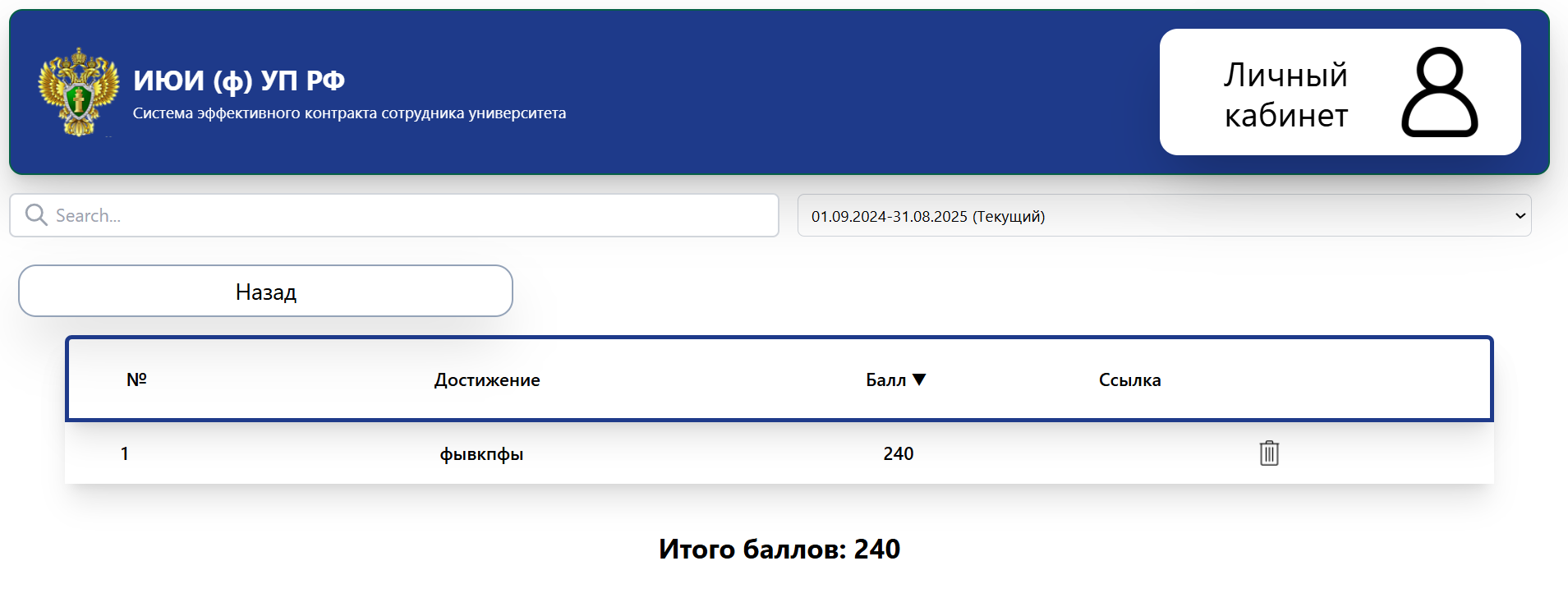


Рисунок 2.5.6 – Макет интерфейса страницы достижений сотрудника

При нажатии на кнопку перехода в личный кабинет происходит переход на страницу ЛК, в котором представлен список показателей с заполненными значениями баллов для авторизованного пользователя. Кнопка «Выйти» позволяет выйти из аккаунта. Поле, которое находиться слева от «Выйти», отображает фамилию текущего авторизованного пользователя. Снизу находиться панель, которая позволят при нажатии на соответствующие кнопку вернуться назад к предыдущей странице, добавить новое достижение либо вывести свои данные в отчет, для роли «Сотрудник» кнопка вывести данные в отчет не будет отображаться. На рисунке 2.5.7 представлен макет интерфейса страницы личного кабинета (списка показателей) авторизованного пользователя:

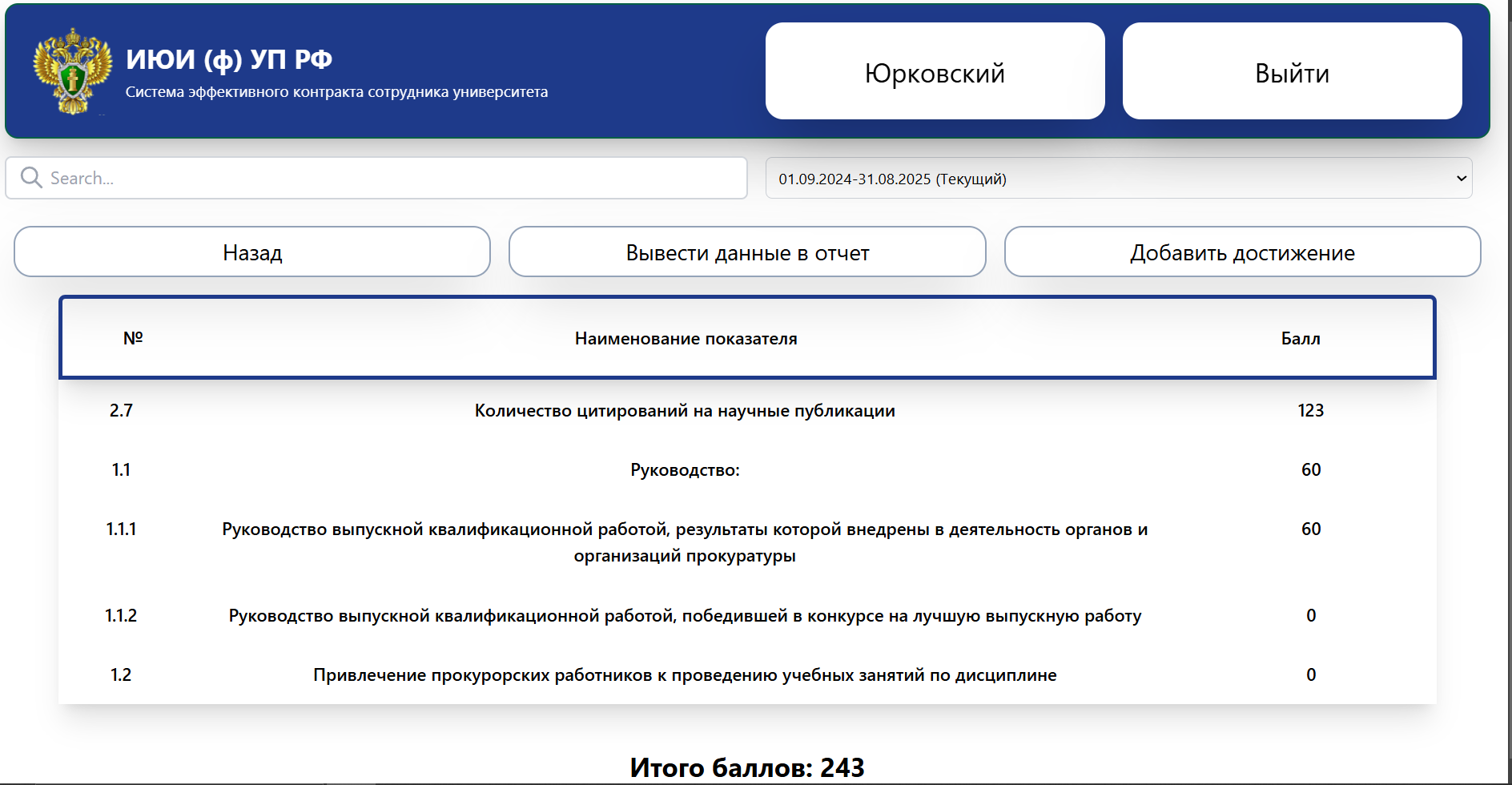


Рисунок 2.5.7 – Макет интерфейса страницы личного кабинета. Список достижений авторизованного пользователя с правами доступа «Сотрудник»

При нажатии на кнопку добавления достижения из страницы ЛК или из страницы списка достижений сотрудника кафедры пользователя с правами доступа «Заведующий кафедрой» происходит открытие модального окна добавления достижения. В нем есть поля выбора показателя, подкатегории, указание подтверждающего документа, подтверждающей ссылки и поля, обозначающего единицу измерения начисления баллов. Снизу представлены итоговый балл достижения и кнопка внесения. В зависимости от типа достижения различаются условия начисления баллов, потому имеется возможность ввести как одно из двух полей подтверждения, так и оба сразу. На рисунке 2.5.8 представлен макет интерфейса модального окна добавления достижения, на рисунке 2.5.9 показано модальное окно добавления с подтверждающим условием ссылкой:

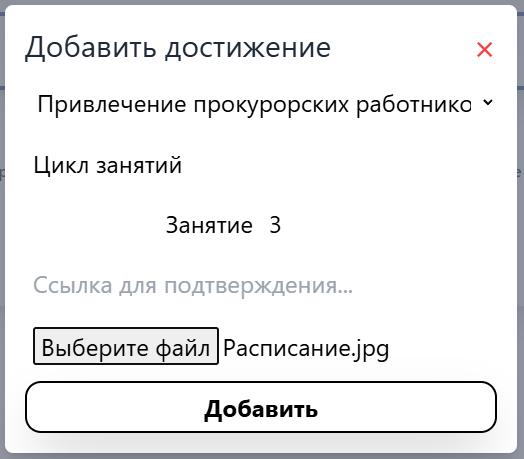


Рисунок 2.5.8 – Макет интерфейса модального окна добавления достижения

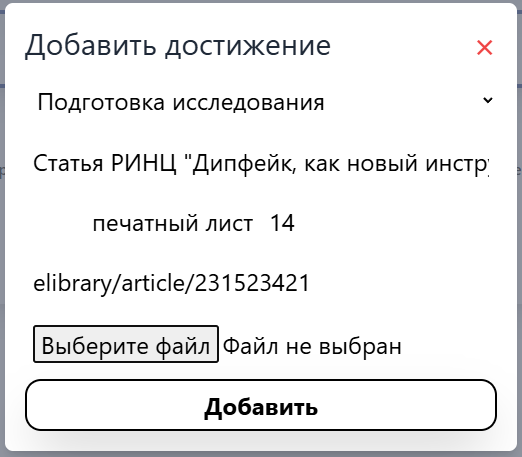
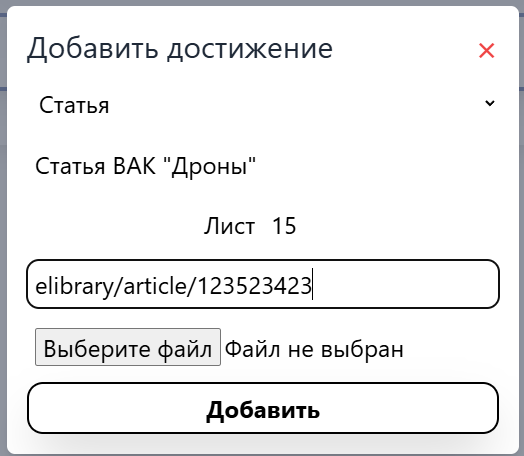


Рисунок 2.5.9 – Макет интерфейса модального окна добавления достижения с ссылкой

При нажатии на любое достижение в списке для пользователя с Ролью «Ответственный за составление рейтинга», а также при нажатии на достижение у сотрудника подразделения, заведующим которого является текущий пользователь с роль «Заведующий кафедрой» достижений из любой страницы со списками достижений можно перейти в окно редактирования детального достижения, на котором можно посмотреть подкатегорию и условие начисления балла, а также скачать и посмотреть его. На рисунке 2.5.10 представлен макет интерфейса страницы редактирования достижения.



Сохранить изменения

Рисунок 2.5.10 – Макет интерфейса модального окна детализированного достижения на редактирование.

## 2.6 Архитектура системы

Для описания архитектуры системы была использована диаграмма, созданная в соответствии с методологией C4 [13]. Данная методология предлагает четыре уровня абстракции, которые позволяют последовательно детализировать структуру системы. Первый уровень, Context, предназначен для визуализации основных взаимодействий системы с внешними элементами. На уровне Containers система разбивается на крупные блоки, представляющие собой контейнеры, которые выполняют определенные функции. Третий уровень, Components, детализирует контейнеры, выделяя в них отдельные компоненты. Наконец, уровень Code обеспечивает максимальную детализацию, где каждый элемент архитектуры соотносится с конкретными строками программного кода, что позволяет разработчикам лучше понимать взаимосвязи и реализацию системы.

Первый уровень контекста (Context), который представлен на рисунке 2.6.1 моделируется для того, чтобы показать связь пользователей ИС и приложения. Все типы пользователей, а именно сотрудник, заведующий кафедрой, ответственный за рейтинг и администратор взаимодействуют с системой, но, также, администратор занимается администрированием ее, т.е. выполняет функции по ее сопровождению и дополнению.

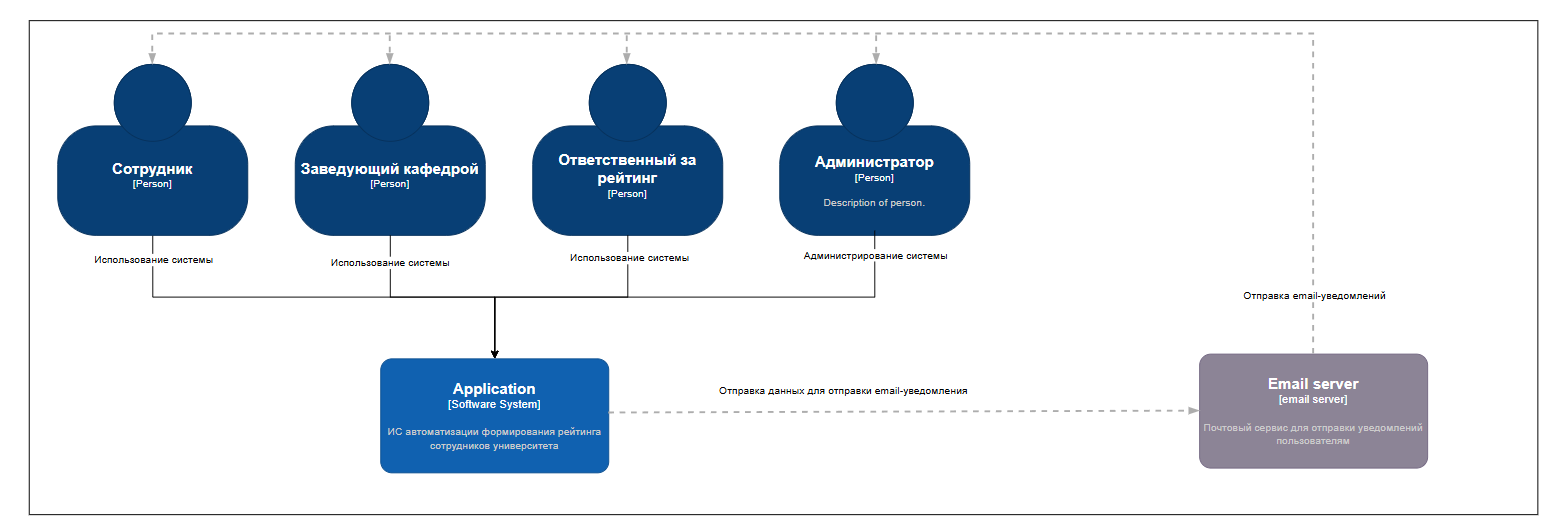


Рисунок 2.6.1 – Уровень 1 контекста (Context) системы

Второй уровень архитектуры, представленный диаграммой контейнеров приложения, который представлен на рисунке 2.6.2, детализирует взаимодействие между подсистемами разрабатываемой системы. Эта диаграмма показывает, что информационная система эффективного контракта сотрудников ИЮИ (ф) УП РФ состоит из нескольких логически связанных контейнеров. Веб-приложение рейтинга университета – клиентская часть ИС, работающая в браузере, которая разработана при помощи фреймворка языка JavaScript – VueJS и выполняющее функцию интерфейса для пользователей ИС эффективного контракта. Оно обеспечивает доступ к функционалу системы через веб-браузер. Backend или серверная часть ИС – это часть системы, которая реализована на фреймворке языка Python – Django. Она выполняет большую часть логических операций во всей ИС, а также отвечает за обработку пришедших данных и взаимодействие с базой данных. DataBase – это база данных, которая реализована при помощи отечественной СУБД PostgreSQL. Она предназначена для хранения всей информации о сущностях, которые используются в системе, а именно, пользователи, подразделения, показатели, достижения преподавателей и сотрудники.

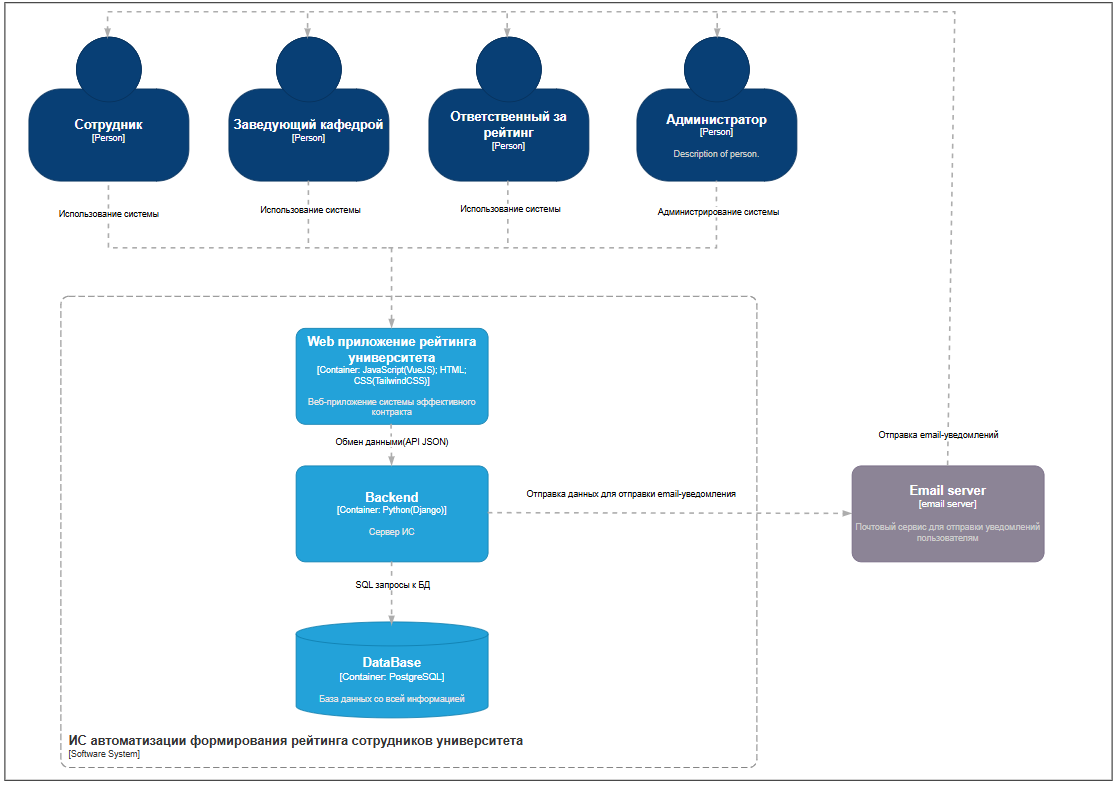


Рисунок 2.6.2 – Уровень 2 контейнеров (Containers) системы

Доступ пользователей к системе осуществляется через веб-интерфейс, который представляет собой сайт, запускаемый в браузере. Клиентское приложение, работающее в браузере, взаимодействует с сервером через API, используя для обмена данными формат JSON. Серверная часть системы обрабатывает запросы, выполняя операции с базой данных, такие как получение, сохранение или модификация данных. Диаграмма контейнеров визуализирует взаимодействие между всеми элементами системы, показывая, как они совместно обеспечивают выполнение требуемых функций.

Третий уровень архитектуры, представленный диаграммой компонентов приложения, которые представлены на рисунках 2.6.3 и 2.6.4, детализирует контейнеры и показывает, как внутри устроены серверная и клиентская части, также показывает их взаимодействие. Все компоненты серверной части разработаны на Python с использованием фреймворка Django, а все компоненты клиентской части реализованы при помощи VueJS.

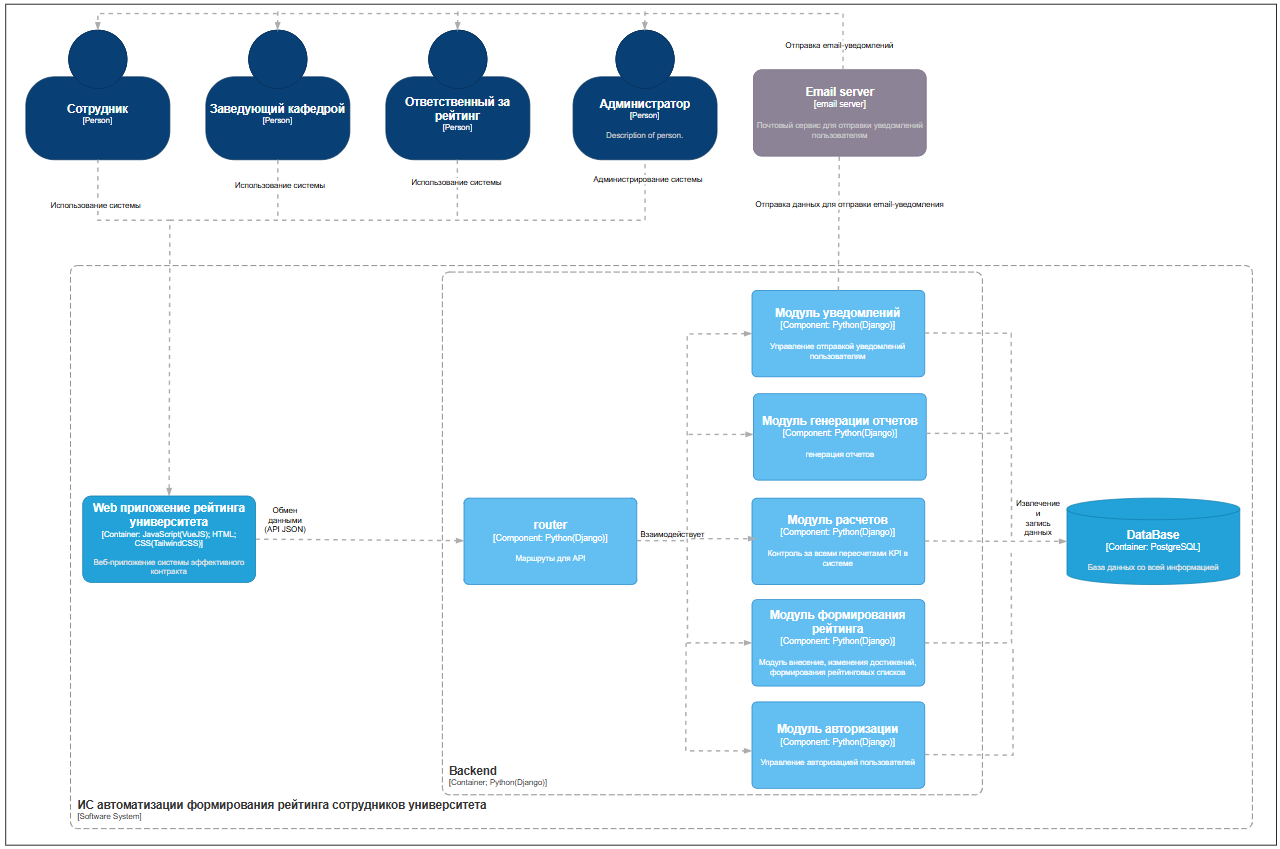


Рисунок 2.6.3 – Уровень 3 компонентов (Components) бэкенда системы

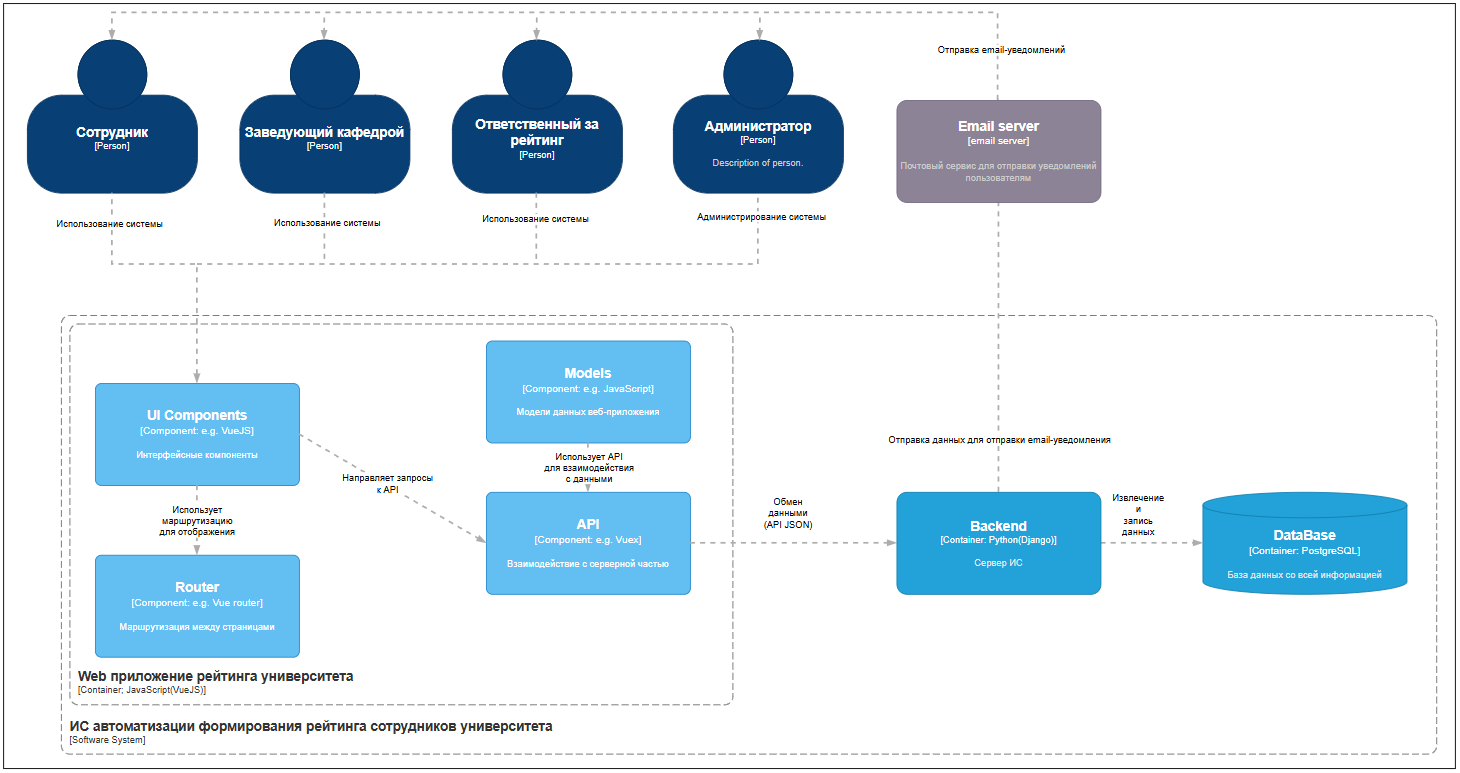


Рисунок 2.6.3 – Уровень 3 компонентов (Components) фронтэнда бэкенда системы

Рассмотрим внутреннюю структуру контейнера **клиентской части** системы. Веб-приложение, которое обеспечивает взаимодействие пользователя с системой через браузер, состоит из нескольких ключевых компонентов, каждый из которых выполняет определенные функции. Основная задача приложения — предоставление пользовательского интерфейса для загрузки файлов, подтверждающих достижения, скачивания подтверждающих документов, а также формирования и загрузки итоговой отчетности. Все компоненты взаимодействуют между собой и с серверной частью системы посредством HTTP-запросов.

1. **Router (Vue Router)**

Компонент маршрутизации, отвечающий за навигацию между страницами веб-приложения. В зависимости от текущего URL, Router определяет, какие компоненты должны быть отображены пользователю, обеспечивая корректное переключение между разделами приложения.

1. **Models**

Набор моделей данных, используемых для хранения и обработки информации, получаемой от сервера через API. Модели служат промежуточным звеном между данными, поступающими извне, и их представлением в интерфейсе.

1. **API (Vue-axios)**

Слой, отвечающий за взаимодействие клиентской части с сервером. Он обеспечивает выполнение HTTP-запросов, таких как загрузка подтверждающих документов, получение итоговой отчетности или скачивание уже загруженных файлов. Этот компонент играет ключевую роль в обмене данными между клиентом и сервером.

1. **UI Components (VueJS)**

Набор интерфейсных компонентов, включая таблицы, шапки, кнопки и другие элементы, которые формируют пользовательский интерфейс. Эти компоненты обеспечивают визуальное представление данных и взаимодействие пользователя с приложением.

Взаимодействие между компонентами происходит следующим образом:

* Пользователь взаимодействует с интерфейсом (UI Components), который отправляет запросы к API для получения или отправки данных.
* Модели (Models) обрабатывают данные, полученные через API, и передают их в интерфейсные компоненты для отображения.
* Маршрутизатор (Router) управляет навигацией, определяя, какие компоненты должны быть отображены в зависимости от текущего URL.

Таким образом, все компоненты клиентской части работают согласованно, обеспечивая удобное и эффективное взаимодействие пользователя с системой.

Теперь рассмотрим внутреннюю структуру **серверной части** приложения. Она состоит из нескольких ключевых модулей, каждый из которых выполняет определенные функции, обеспечивая работу системы в целом.

1. **Модуль авторизации**

Этот компонент отвечает за аутентификацию и авторизацию пользователей системы. Он проверяет учетные данные пользователей и предоставляет доступ к функционалу системы в соответствии с их правами и ролями.

1. **Модуль расчетов**

Данный компонент автоматизирует процесс расчета KPI (ключевых показателей эффективности). Он полностью исключает необходимость ручного взаимодействия пользователя с числовыми данными. Пользователю достаточно внести информацию о своих достижениях, а модуль самостоятельно выполнит все необходимые вычисления.

1. **Модуль генерации отчетных документов**

Этот компонент отвечает за автоматическое создание отчетов на основе заранее заложенных шаблонов и данных, хранящихся в системе. Он использует информацию о сотрудниках, подразделениях, показателях и достижениях преподавателей, формируя итоговые документы в соответствии с требованиями.

1. **Модуль формирования рейтинга**

Компонент предназначен для фильтрации и обработки данных, поступающих из базы данных, с целью формирования рейтингов. Например, он позволяет отображать рейтинг сотрудников конкретного подразделения или других групп, определенных пользователем.

1. **Роутер**

Этот компонент отвечает за маршрутизацию запросов API между различными модулями серверной части. Он обеспечивает корректную передачу данных и команд между компонентами системы, взаимодействуя со всеми остальными модулями и гарантируя их слаженную работу.

Таким образом, диаграмма компонентов серверной части наглядно демонстрирует, как взаимодействуют между собой различные модули, обеспечивая выполнение всех необходимых функций системы. Каждый модуль играет важную роль в обработке данных, автоматизации процессов и предоставлении пользователям доступа к нужной информации.

1. **Модуль уведомлений**

Компонент предназначен для формирования текстов уведомлений, которые будут перенаправлены на сервер электронной почты для дальнейшей отправки пользователям.

## 2.7 Инструменты разработки

Для успешного проектирования и разработки ИС необходимо выбрать набор инструментов, который позволит правильно распределить ресурсы и сделать разработку и саму ИС гибкими и эффективными, а также позволяющие будущему продукту быть способным выполнять все необходимые функциональные возможности.

В этом разделе будет описан весь набор программных инструментов, которые были использованы при разработке и проектировании продукта, а также аргументация их выбора.

Для разработки логической модели хранилища данных был выбран инструмент Draw.io [8]. Это бесплатное веб-приложение, предназначенное для создания различных типов диаграмм, включая рабочие процессы, организационные и сетевые схемы. Сервис отличается интуитивно понятным интерфейсом и поддерживает экспорт проектов в популярные форматы, такие как JPG, PNG и другие. Именно благодаря этим характеристикам данный инструмент был выбран для выполнения задачи.

Для построения моделей в нотации IDEF0, которые описывали бизнес-процессы, а также модели потоков данных в нотации DFD был использован специализированный редактор RAMUS [9], который направлен на работу исключительно с двумя описанными нотациями, предоставляя максимальное удобство, шаблоны, а также помогая сохранять стандарты нотаций.

Архитектура системы была описана с использованием модели C4. Этот подход был выбран благодаря его возможности наглядно демонстрировать структуру системы на разных уровнях абстракции. Модель C4 обеспечивает простоту восприятия, что делает ее удобной как для технических специалистов, так и для пользователей, не обладающих глубокими знаниями в области разработки программного обеспечения.

Для работы приложения необходимо обеспечить работу и взаимодействие бекэнда, фронтэнда, СУБД, для хранения табличных (объектных) данных, а также необходимо обеспечить тестовый сервер (хостинг) и сервер (хостинг), на котором ИС будет запущена в работу окончательно включая доменное имя и выделить на сервере место, для хранения медиафайлов, документов подтверждений.

Был выбран веб-браузер в качестве клиентского приложения с целью обеспечить доступность продукта из любой точки мира с подключением к интернету. Этот выбор обеспечивает гибкость использования и удобство для конечных пользователей.

Для разработки программы было принято решение использовать интегрированную среду разработки (IDE) Visual Studio Сode (VSCode). Выбор был сделан исходя из простоты и легковесности самого приложения, широкой возможности модификации функционала персонально под каждого разработчика благодаря огромному количеству библиотек.

Для бэкенда было решено остановиться на Python вместе с фреймворком Django [10]. Python известен своей простотой и эффективностью в разработке, а Django предоставляет обширный набор инструментов для быстрой создания веб-приложений, обеспечивая высокую производительность и надежность.

JavaScript + Vue [11]: для фронтэнда был использован JavaScript в сочетании с Vue. предоставляет простой и интуитивно понятный способ создания пользовательского интерфейса, обеспечивая отзывчивость и удобство использования, а JavaScript позволяет работать с динамическими типами данных.

Для управления базой данных в рамках РФ было требованием использовать PostgreSQL [12] в качестве системы управления базами данных (СУБД). PostgreSQL является надежной и производительной СУБД, которая хорошо интегрируется с VSСode, что упрощает работу с базой данных, а также postgres активно применяется во всех сферах в РФ, как замена иностранному ПО.

Тестовый сервер (хостинг): необходим для тестирования работоспособности ИС, выявления ошибок и замечаний к интерфейсу и работе системы и их дальнейшее устранение. Должен быть платным, но не дорогим, для избегания высоких затрат на тестирование.

Рабочий сервер (хостинг): в рамках хостинга будут развернуты и бек, и фронт части проекта, также, как и СУБД. На хостинге должно быть определено место для хранения медиафайлов (документы, PDF, картинки). Обеспечить достаточный для хранения большого количества файлов объем места, для хранения данных о ежеквартальных рейтингах на ближайшие 5 лет. Минимальный объем диска должен составлять примерно 2ТБ. ИС должна быть развернута на мощностях Университета.

Доменное имя: в целях удобства эксплуатации ИС заказчиком, необходимо обеспечить простое и понятное доменное имя. Уровень домена должен быть 3м (например rating.iagprf.ru).

Выбор данных инструментов связан с их эффективностью, а также наличием опыта использования данных инструментов разработчиком.

В качестве аккаунта рассылки сообщений был выбран сервис Яндекс.Почта, в первую очередь из-за того, что это российский сервис, но также потому, что он предоставляет все необходимые возможности для рассылки.

## Заключение по разделу

В разделе проектирования информационной системы были разработаны ключевые модели и схемы, необходимые для реализации системы автоматизации формирования рейтинга сотрудников. Основное внимание было уделено процессу формирования рейтинга с применением проектируемой ИС, что позволило значительно сократить ручной труд и автоматизировать 75% процессов, связанных с расчетом KPI и формированием отчетов.

Была разработана схема потоков данных (DFD), которая наглядно демонстрирует, как данные будут перемещаться между различными компонентами системы, включая базу данных, клиентскую и серверную части. Это позволило устранить необходимость документооборота между этапами процесса и централизовать хранение данных.

Также была решена задача унификации показателей, что позволило создать модель данных, которая охватывает все возможные типы достижений сотрудников. На основе этой модели была разработана инфологическая и физическая модель базы данных, которая обеспечивает целостность и эффективность хранения информации. В качестве СУБД был выбран PostgreSQL, что соответствует требованиям заказчика и обеспечивает надежность и производительность системы.

Были разработаны макеты интерфейса, которые обеспечивают удобство использования системы для всех категорий пользователей, включая сотрудников, заведующих кафедрами, ответственных за составление рейтинга и администраторов. Интерфейс системы интуитивно понятен и позволяет легко сортировать, фильтровать и просматривать данные.

В качестве инструментов разработки были выбраны современные технологии, которые будут обеспечивать гибкость, производительность и возможность дальнейшего развития системы.

Итогом проектирования стала детализированная архитектура системы, которая включает в себя все необходимые компоненты для успешной реализации проекта.

Таким образом, в результате проектирования была создана основа для разработки автоматизированной системы, которая позволит значительно упростить процесс формирования рейтинга сотрудников и повысить эффективность работы университета.

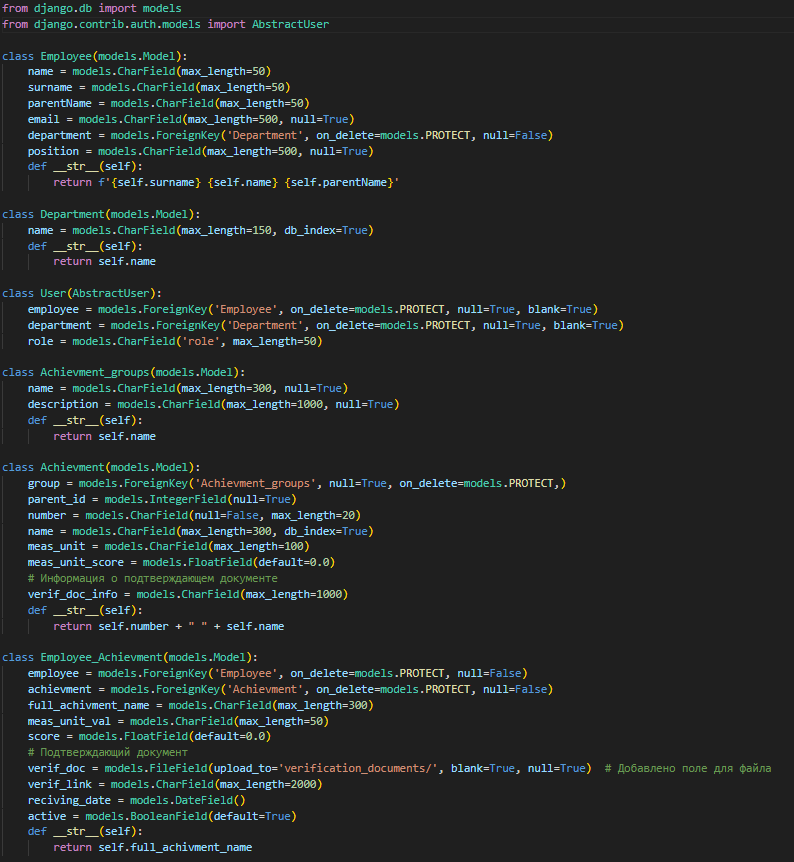
# 3 Реализация проекта

## 3.1 Реализация базы данных

База данных системы была разработана с использованием СУБД PostgreSQL версии 16 [12]. Для управления структурой базы данных и её изменениями применялась модель данных, описанная в Django. Эта модель позволяет автоматически генерировать структуру базы данных и обеспечивает её корректную работу. На рисунке 3.1.1 представлена схема данной модели.

После описания модели данных выполняются миграции. Миграции — это механизм, который создает таблицы в базе данных, изменяет их структуру, добавляет новые поля и устанавливает связи между таблицами с использованием внешних ключей (foreign key). Важным преимуществом этого подхода является возможность выполнения SQL-операций через язык Python, что исключает необходимость написания SQL-запросов вручную.

Данный метод был выбран для обеспечения удобства и скорости создания таблиц, а также для упрощения управления структурой базы данных на этапах разработки и развертывания приложения. Кроме того, миграции позволяют сохранять историю изменений схемы базы данных, что упрощает внесение правок и отслеживание эволюции структуры БД.

 Рисунок 3.1.1 – Модель данных, описанная при помощи python

На стартовом этапе разработки была выполнена первичная миграция, которая задала базовую структуру всех таблиц в базе данных. Эта миграция играет ключевую роль в формировании БД, обеспечивая возможность добавления, хранения и обработки всех данных, необходимых для функционирования системы. Среди таких данных выделяются проекты, вопросы и другие важные сущности.

Для удобства тестирования и проверки работоспособности системы база данных была предварительно заполнена тестовыми данными. В их состав вошли сотрудники, пользовательские аккаунты для сотрудников с разными ролями, аккаунт администратора, несколько подразделений, небольшой список показателей, а также нескольким сотрудникам были присвоены достижения. Использование тестовых данных позволило оперативно проверить корректность создания таблиц, установления связей между ними (включая внешние ключи) и выполнения запросов в приложении.

Таким образом, первичная миграция определяет минимально необходимую структуру базы данных, которая обеспечивает хранение и обработку всех ключевых данных, требуемых для полноценной работы системы.

## 3.2 Реализация API-запросов

Для реализации всех функций логики ИС, функций пересчета KPI, фильтрации и сортировки данных, для вывода ее пользователям, а также для внесения в базу данных новых, сформированных в типовые JSON-массивы, данных о достижениях были реализованные API на фреймворке языка python – Django. Представленные на рисунке 3.2.1 URL-адреса обрабатывают запросы с клиентской части (фронтэнда) и взаимодействуют с базой данных, получая, внося и редактируя оттуда информацию.



Рисунок 3.2.1 – URL-адреса API запросов, реализованных в проекте на Django

Каждый из API запросов берет на себя определенную логику, которая отвечает за работоспособность ИС. От этих URL-адресов клиент получает ответ, которые зависит от успешности выполнения запроса к базе данных со стороны бэкенда. Ниже представлена таблица 3.2.1 с основными возможными ответами, данными, кодами и параметрами, которые могут передаваться в запросе:

Таблица 3.2.1 – Возможные ответы от API запросов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL** | **Ответ** | **Тип данных** | **Описание** | **Параметры запроса** |
| api/employee/<employee\_id>/achievement/<achievment\_id>/achievements/ | 200 OK  {  "employee": "Юрковский",  "achievment": "Руководство вкр",  "achievements": [  {  "id": 11,  "employee\_id": 1,  "achievment\_name": "Руководство вкр",  "full\_name": "Выпуск 2024",  "score": 80.0,  "meas\_unit\_val": "4",  "verif\_doc": "/verification | JSON Array | Список достижений конкретного сотрудника по конкретному показателю | /<achievment\_id> - идентификатор достижения |
| 404 Not found  {"error": "Достижения не найдены для данного сотрудника и достижения"} | - |

Продолжение таблицы 3.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| api/v1/employe\_achievments/employee/<int:employee\_id> | 200 OK  {  "employee": "Юрковский",  "achievements": [  {  "id": 3,  "number": "1.1",  "achievment\_name": "Выполнение объема работы учебной нагрузки",  "score": 0,  "meas\_unit\_val": null,  "verif\_doc": null  },  {  "id": 4,  "number": "1.2",  "achievment\_name": "Руководство вкр",  "score": 80.0,  "meas\_unit\_val": null,  "verif\_doc": null | JSON Array | Список показателей с заполненными значениями KPI для конретного сотрудника | / <int: employee\_id> - идентификатор сотрудника |
| 404 Not found  {"error": "Достижения не найдены для данного сотрудника и достижения"} | - |

Продолжение таблицы 3.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| api/v1/employee\_achievements/teachers/<int:employee\_id>/delete | 200 OK  {"success": "Достижения удалено"} | JSON Array | Удаление достижения | /<int:employee\_id> - идентификатор сотрудника |
| 404 Not found  {"error": "Достижения не найдены для данного сотрудника и достижения"} | - |
| api/v1/employee\_achievements/teachers/<int:employee\_id>/ | 200 OK  {"success": "Достижения добавлено"} | JSON Array | Добавление нового достижения для сотрудника | /<int:employee\_id> - идентификатор сотрудника |
| 404 Not found  {"error": "Достижения не найдены для данного сотрудника и достижения"} | - |
| api/v1/departments/<int:department\_id>/teachers/ | 200 OK  [  {  "id": 1,  "surname": "Юрковский",  "name": "Егор",  "parentName": "Алексеевич",  "total\_score": 180.0}  ] | JSON Array | Список сотрудников подразделения с заполненными значениями KPI | /<int:department\_id> - идентификатор подразделения |
| 404 Not found  {"error": "Сотрудники не найдены для данного подразделения"} | - |
| admin/ | 200 OK | JSON Array | Панель Администратора | - |
|  | 401 unauthorized | - | - |

## 3.3 Отправка уведомлений на почту

Важной функциональной возможностью разрабатываемой информационной системы является автоматическая отправка уведомлений на электронную почту пользователей при изменении или удалении достижений. Это позволяет сотрудникам оперативно получать информацию о статусе своих достижений и своевременно вносить необходимые корректировки. Данный функционал реализован с использованием библиотеки yagmail, которая обеспечивает простую и надежную отправку электронных писем через SMTP-сервер.

Отправка уведомлений реализована в рамках API-запроса на удаление достижения сотрудника. При удалении достижения система автоматически отправляет уведомление на электронную почту сотрудника, указанную в запросе. Уведомление содержит информацию о названии удаленного достижения и причину его удаления. Это позволяет сотруднику понять, какие именно данные были удалены и что необходимо исправить для повторного внесения достижения.

Пример реализации данного функционала представлен в классе DeleteEmployeeAchievementApiView, который наследуется от APIView фреймворка Django. В этом классе реализован метод delete, который выполняет следующие действия:

Из тела запроса извлекаются электронная почта сотрудника и причина удаления достижения. Эти данные необходимы для формирования и отправки уведомления. Если электронная почта указана, система формирует и отправляет письмо с использованием библиотеки yagmail. Письмо содержит информацию о названии удаленного достижения и причину его удаления. Это позволяет сотруднику оперативно получить информацию о внесенных изменениях и при необходимости внести корректировки. После успешной отправки уведомления система удаляет достижение из базы данных. В случае успешного выполнения операции система возвращает ответ с кодом статуса 204 (No Content), что означает успешное удаление данных.

Пример кода, реализующего данный функционал представлен в листинге 3.3.1:

Листинг 3.3.1 – Отправка уведомлений в рамках удаления достижения

|  |
| --- |
| class DeleteEmployeeAchievementApiView(APIView):  """Удаляет конкретное достижение сотрудника по его ID"""  def delete(self, request, achievement\_id):  achievement = get\_object\_or\_404(Employee\_Achievment, id=achievement\_id)    # Получаем email из тела запроса  email = request.data.get('email')  reason = request.data.get('reason')    if email:  # Отправка сообщения об удалении  yag = yagmail.SMTP('rezervdesu15@gmail.com', 'ddvd jduu prdi ktih')  yag.send(  email, # Используем email из запроса  f"Удаление достижения {achievement.full\_achivment\_name}",  f"Достижение {achievement.full\_achivment\_name} было удалено ввиду его некорректности, исправьте ошибки и внесите корректное достижение\nПричина удаления: {reason}"  )    achievement.delete()  return Response({'message': 'Достижение удалено!'}, status=status.HTTP\_204\_NO\_CONTENT) |

Помимо отправки уведомлений при удалении достижений, система также поддерживает отправку уведомлений при обновлении данных. Это позволяет сотрудникам быть в курсе любых изменений, внесенных в их достижения. Например, если ответственный за рейтинг или заведующий кафедрой вносит изменения в данные о достижениях сотрудника, система автоматически отправляет уведомление на электронную почту сотрудника с информацией о внесенных изменениях.

Реализация отправки уведомлений при обновлении данных аналогична отправке уведомлений при удалении. В API-запросе на обновление данных система также извлекает электронную почту сотрудника и формирует письмо с информацией о внесенных изменениях.

**Преимущества автоматической отправки уведомлений:**

Автоматическая отправка уведомлений на электронную почту имеет ряд преимуществ:

* Сотрудники получают информацию о внесенных изменениях в режиме реального времени, что позволяет им оперативно реагировать на любые изменения в своих данных.
* Уведомления содержат подробную информацию о внесенных изменениях, что делает процесс формирования рейтинга более прозрачным и понятным для всех участников.
* Автоматическая отправка уведомлений исключает необходимость ручного информирования сотрудников о внесенных изменениях, что снижает нагрузку на ответственных за формирование рейтинга.
* Уведомления отправляются на электронную почту, что делает процесс получения информации удобным и доступным для всех сотрудников.

Исходя из описанного, реализация автоматической отправки уведомлений на электронную почту является важным элементом разработанной ИС. Она позволяет повысить оперативность и прозрачность процесса формирования рейтинга, а также снизить нагрузку на ответственных за его составление.

## 3.4 Описание клиентской части

Клиентская часть приложения представляет собой динамическое веб-приложение, разработанное с использованием фреймворка Vue.js 3. Данный фреймворк выбран благодаря его высокой производительности, компонентному подходу и возможностям эффективного управления состоянием приложения.

**Архитектура клиентской части**

Клиентская часть приложения построена на основе компонентного подхода, который является одной из ключевых концепций Vue.js. Каждый компонент представляет собой независимый блок, который отвечает за отображение определенной части интерфейса и взаимодействие с пользователем. Такой подход позволяет легко поддерживать и расширять приложение, а также повторно использовать компоненты в разных частях системы.

В системе широко используются таблицы для представления данных о сотрудниках, подразделениях, достижениях, что делает работу с информацией интуитивно понятной и удобной для пользователей. Для обеспечения гибкости и повторного использования кода были созданы универсальные компоненты таблицы, позволяющие адаптировать отображение данных для различных сущностей без дублирования логики.

Основные компоненты клиентской части включают:

HeaderBlock — компонент, отвечающий за отображение шапки приложения. Он содержит навигационные элементы, такие как кнопки для перехода в личный кабинет и выхода из системы.

BaseTable — универсальный компонент для отображения табличных данных. Он поддерживает сортировку, фильтрацию и пагинацию, что делает его удобным для работы с большими объемами информации.

TableRow и TableColumn — вспомогательные компоненты, используемые для построения строк и столбцов таблицы. Они обеспечивают гибкость в настройке внешнего вида и поведения таблиц.

Модальные окна — используются для добавления и редактирования данных, таких как достижения сотрудников. Модальные окна реализованы с использованием реактивных свойств Vue.js, что позволяет динамически управлять их состоянием (открытие/закрытие).

Основными компонентами, формирующими таблицу, являются BaseTable.vue, TableRow.vue и TableColumn.vue. Эти компоненты совместно обеспечивают отображение, сортировку, фильтрацию и взаимодействие с данными, загруженными с сервера.

Компонент BaseTable.vue является основным контейнером для представления таблицы. Он отвечает за загрузку, отображение и управление данными, передаваемыми в таблицу. В этом компоненте реализована логика обработки сортировки и фильтрации данных, а также механизмы взаимодействия с серверной частью для загрузки информации постранично. BaseTable.vue содержит шаблон, определяющий структуру таблицы, и использует дочерние компоненты TableRow.vue и TableColumn.vue для отображения строк и отдельных ячеек.

Компонент TableRow.vue представляет собой отдельную строку таблицы и используется внутри BaseTable.vue. Он получает данные в виде объекта и отображает соответствующую информацию в колонках. В данном компоненте могут быть реализованы дополнительные элементы управления, такие как кнопки редактирования или удаления записи, а также визуальные индикаторы статуса данных.

Компонент TableColumn.vue отвечает за отображение конкретного столбца внутри строки. Он может поддерживать различные форматы данных, включая текстовые значения, числовые данные, даты и пользовательские компоненты, такие как кнопки или выпадающие списки. Этот компонент используется внутри TableRow.vue и предоставляет гибкую настройку отображаемой информации в зависимости от переданных параметров.

**Маршрутизация и навигация**

Для организации навигации между страницами приложения используется Vue Router. Маршруты определены для каждой страницы, включая личный кабинет, страницу рейтинга сотрудников, страницу рейтинга кафедр и страницу с подробной информацией о достижениях. Vue Router обеспечивает плавные переходы между страницами без необходимости полной перезагрузки, что улучшает пользовательский опыт.

Маршруты клиентской части:

* / и /teachersRatingPage: Главная страница, на которой отображается рейтинг преподавателей. Компонент teachersRatingPage.vue отвечает за отображение этой страницы.
* /departamentRatingPage: Страница с рейтингом подразделений. Компонент departamentRatingPage.vue отвечает за отображение этой страницы.
* /department/:id: Страница с детальной информацией о конкретном подразделении. Компонент depatmentInsidePage.vue отвечает за отображение этой страницы.
* /teachers/:id: Страница с детальной информацией о конкретном преподавателе. Компонент TeachersInsidePage.vue отвечает за отображение этой страницы.
* /login: Страница авторизации. Компонент loginPage.vue отвечает за отображение формы входа.
* /profile: Личный кабинет пользователя. Компонент profilePage.vue отвечает за отображение этой страницы.
* /profile/addAchievements: Страница добавления достижений. Компонент addAchievements.vue отвечает за отображение этой страницы.
* /employee/:empl\_id/achievment/:ach\_id: Страница с детальной информацией о конкретном достижении преподавателя. Компонент achievmentDetailed.vue отвечает за отображение этой страницы.

**Работа с состоянием и API**

Для управления состоянием приложения и взаимодействия с серверной частью используется библиотека Axios, которая позволяет выполнять HTTP-запросы к API. Все запросы к серверу, такие как получение данных о достижениях, сотрудниках и кафедрах, выполняются асинхронно, что обеспечивает плавную работу интерфейса без блокировки пользовательского взаимодействия.

Пример запроса к API для получения данных о достижениях сотрудника представлен в листинге 3.4.1:

Листинг 3.4.1 – Запрос для получения данных о достижениях сотруднипка

|  |
| --- |
| const achievementsResponse = await axios.get(  `http://127.0.0.1:8000/api/v1/employe\_achievments/employee/${employeeId}`  );  achivmentsData.value = achievementsResponse.data.achievements; |

Для управления состоянием приложения, таким как данные пользователя, токен авторизации и роль, используются реактивные переменные Vue.js (ref и reactive). Это позволяет динамически обновлять интерфейс при изменении данных.

**Обработка ошибок и уведомления**

В приложении реализована система уведомлений, которая информирует пользователя о результатах выполнения операций, таких как добавление или удаление достижений. Уведомления отображаются в виде всплывающих сообщений и автоматически исчезают через несколько секунд. Для обработки ошибок, таких как сбои при запросах к API, используется блок try-catch, который позволяет корректно обрабатывать исключения и информировать пользователя о возникших проблемах.

Пример обработки ошибки при удалении достижения представлен в листинге 3.4.2:

Листинг 3.4.2 – обработка ошибки при удалении достижения

|  |
| --- |
| try {  const response = await axios.delete(  `http://127.0.0.1:8000/api/v1/delete\_employee\_achievement/${achievementId}/`,  {  headers: { Authorization: `Token ${token.value}` }  }  );  successMessage.value = 'Успешно удалено достижение!';  } catch (error) {  errorMessage.value = 'Ошибка при удалении достижения. Попробуйте позже.';  } |

**Адаптивный дизайн и UX**

Клиентская часть приложения разработана с учетом принципов адаптивного дизайна, что позволяет корректно отображать интерфейс на устройствах с различными разрешениями экрана. Для создания адаптивных layouts используются CSS-фреймворки, такие как Tailwind CSS, который предоставляет готовые классы для быстрой настройки отступов, размеров и других стилей.

Пример использования Tailwind CSS для создания адаптивного layout представлен в листинге 3.4.3:

Листинг 3.4.3 – Использования Tailwind CSS для создания адаптивного layout

|  |
| --- |
| <div class="grid grid-cols-2 items-center mt-4 mx-4">  <div class="relative">  <input  class="appearance-none border-2 pl-10 border-gray-300 hover:border-gray-400 transition-colors rounded-md w-full py-2 px-3 text-gray-800 leading-tight focus:outline-none focus:ring-blue-900 focus:border-blue-900 focus:shadow-outline"  type="text"  placeholder="Search..."  />  </div>  </div> |

Клиентская часть приложения представляет собой современное веб-решение, которое сочетает в себе высокую производительность, удобство использования и гибкость. Использование Vue.js и других современных технологий позволяет создавать интуитивно понятный и отзывчивый интерфейс, который удовлетворяет потребности пользователей и обеспечивает эффективное взаимодействие с системой. Клиентская часть приложения организована модульно, что позволяет легко расширять и модифицировать её в будущем. Гибкая маршрутизация позволяет пользователям быстро находить необходимую информацию, переходить между страницами и взаимодействовать с системой без задержек, что значительно улучшает пользовательский опыт. Вся структура приложения ориентирована на удобство работы с большими объемами данных, обеспечивая высокую производительность и простоту в поддержке.

## 3.5 Модуль генерации отчетов

Модуль генерации отчетов является важной частью информационной системы, так как он позволяет автоматически формировать отчеты в формате Excel и PDF, что значительно упрощает процесс подготовки документации для руководства университета. В текущей версии системы реализована возможность генерации персональных отчетов по достижениям сотрудников, а также отчетов по подразделениям.

Персональный отчет содержит информацию о достижениях конкретного сотрудника, включая наименование показателей, единицы измерения, количество баллов и общий итог по каждому показателю. Отчет формируется на основе данных, хранящихся в базе данных, и выгружается в формате Excel.

Для реализации этой функциональности используется библиотека openpyxl, которая позволяет работать с Excel-файлами в Python. Пример сгенерированного отчета представлен на рисунке 3.5.1

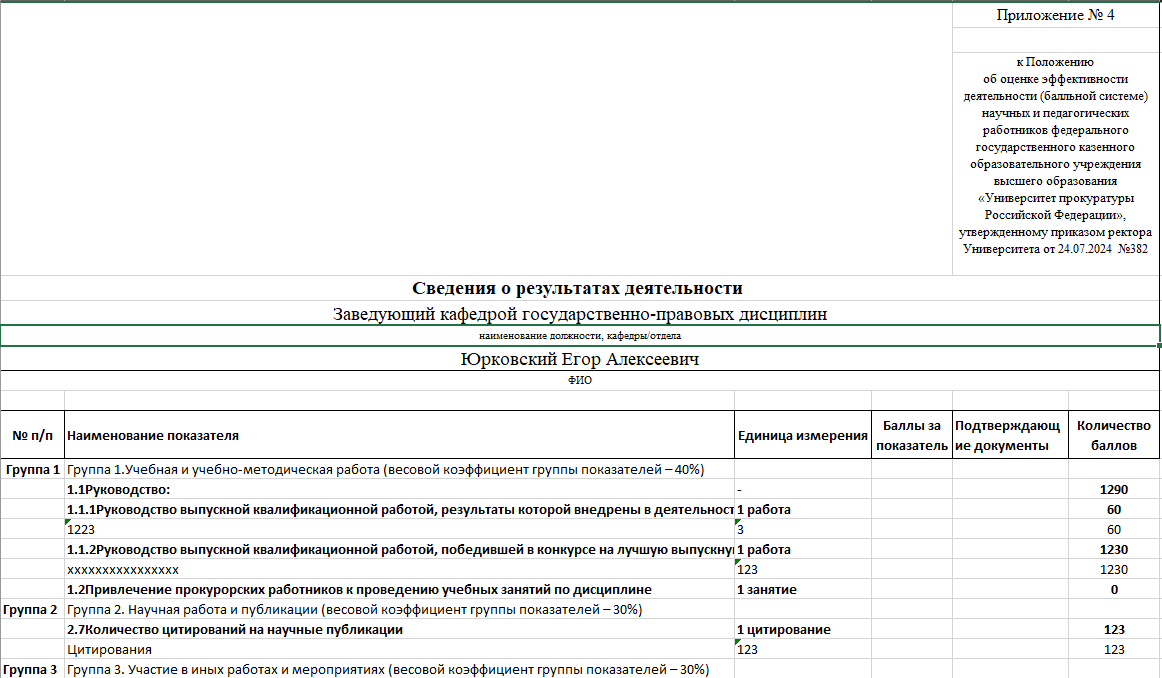


Рисунок 3.5.1 Сгенерированный персональный отчет

Отчет по подразделению содержит сводную информацию о достижениях всех сотрудников, относящихся к данному подразделению. В отчете указываются общие баллы по каждому показателю, а также итоговый рейтинг подразделения.

Аналогично персональному отчету, отчет по подразделению формируется на основе данных из базы данных и выгружается в формате Excel. Для этого также используется библиотека openpyxl. Пример сгенерированного отчета по подразделению представлен на рисунке 3.5.2.

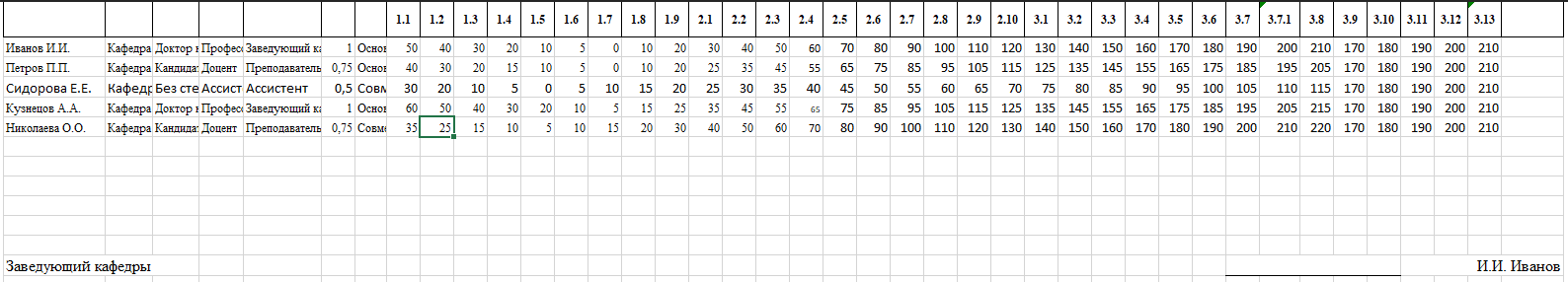


Рисунок 3.5.2 Сгенерированный отчет по подразделению

## 3.6 Описание возможностей текущей версии

На текущий момент разработанная информационная система автоматизации формирования рейтинга сотрудников в рамках «Системы эффективного контракта» представляет собой полнофункциональное веб-приложение, которое позволяет автоматизировать процесс расчета KPI, формирования рейтингов и генерации отчетности. Система обеспечивает следующие ключевые возможности:

* Автоматический расчет KPI: Система автоматически рассчитывает ключевые показатели эффективности сотрудников на основе введенных данных о достижениях. Это исключает необходимость ручного подсчета баллов, что значительно снижает вероятность ошибок и экономит время ответственных лиц;
* Централизованное хранение данных: Все данные о сотрудниках, их достижениях, подразделениях и показателях хранятся в единой базе данных, что обеспечивает удобство доступа к информации и исключает необходимость документооборота между этапами процесса;
* Генерация отчетов: Система поддерживает генерацию отчетов в форматах Excel и PDF, что позволяет быстро формировать необходимую документацию для руководства университета. Отчеты создаются на основе заранее заложенных шаблонов, что обеспечивает их соответствие требованиям;
* Удобный интерфейс: Пользовательский интерфейс системы интуитивно понятен и удобен для всех категорий пользователей. Он включает в себя таблицы с возможностью сортировки и фильтрации данных, что позволяет быстро находить нужную информацию;
* Разграничение прав доступа: Система поддерживает различные роли пользователей (сотрудник, заведующий кафедрой, ответственный за рейтинг, администратор), что обеспечивает безопасность данных и ограничивает доступ к функционалу в зависимости от роли;
* Поддержка различных типов подтверждающих документов: Система позволяет загружать подтверждающие документы в различных форматах (PDF, DOC, JPEG, PNG и др.), что упрощает процесс внесения данных о достижениях;
* Возможность редактирования и удаления данных: Пользователи с соответствующими правами могут редактировать и удалять данные о достижениях, что обеспечивает гибкость в работе с информацией;
* Поиск и фильтрация данных: Система поддерживает поиск и фильтрацию данных по различным критериям, что позволяет быстро находить нужную информацию и анализировать рейтинги сотрудников и подразделений;
* Возможность отправки уведомлений пользователю на электронную почту, если его достижение было удалено или изменено.

Таким образом, текущая версия системы уже обладает всеми необходимыми функциями для автоматизации процесса формирования рейтинга сотрудников. Однако, как и любое программное обеспечение, система имеет потенциал для дальнейшего развития и улучшения.

## 3.7 Пример работы с приложением

Для демонстрации функциональности разработанной информационной системы рассмотрим последовательность работы с приложением на примере пользователя с ролью «Сотрудник».

После успешной авторизации (рисунок 3.7.1) в системе, сотрудник попадает на свою страницу профиля (рисунок 3.7.2). В личном кабинете отображается список всех показателей, по которым сотрудник может внести свои достижения.

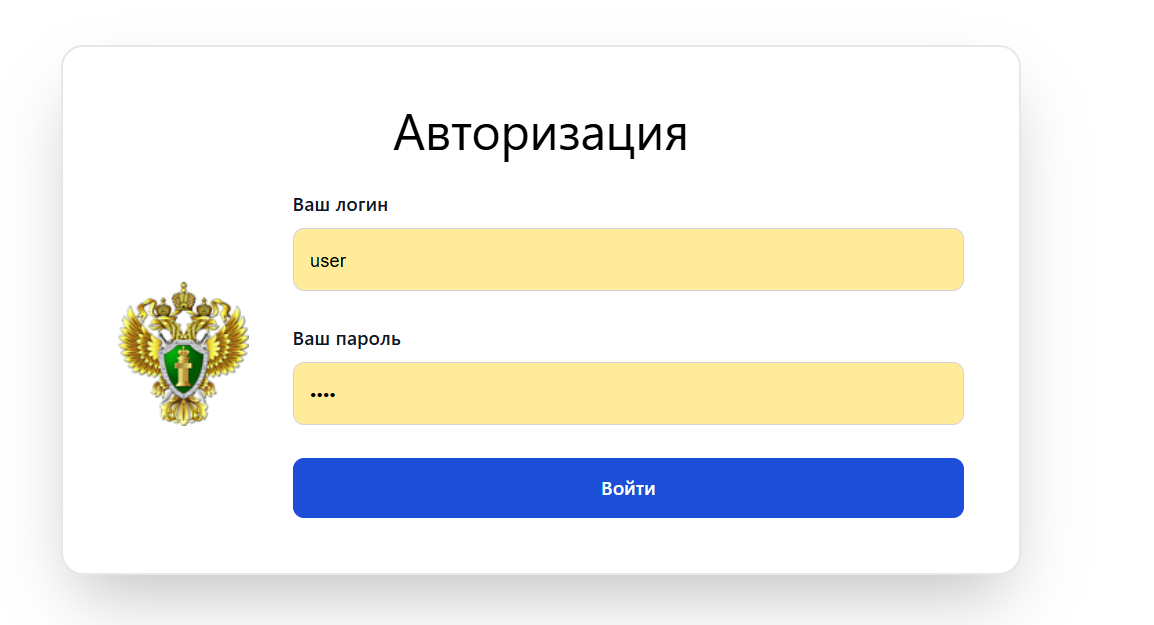


Рисунок 3.7.1 Страница авторизации

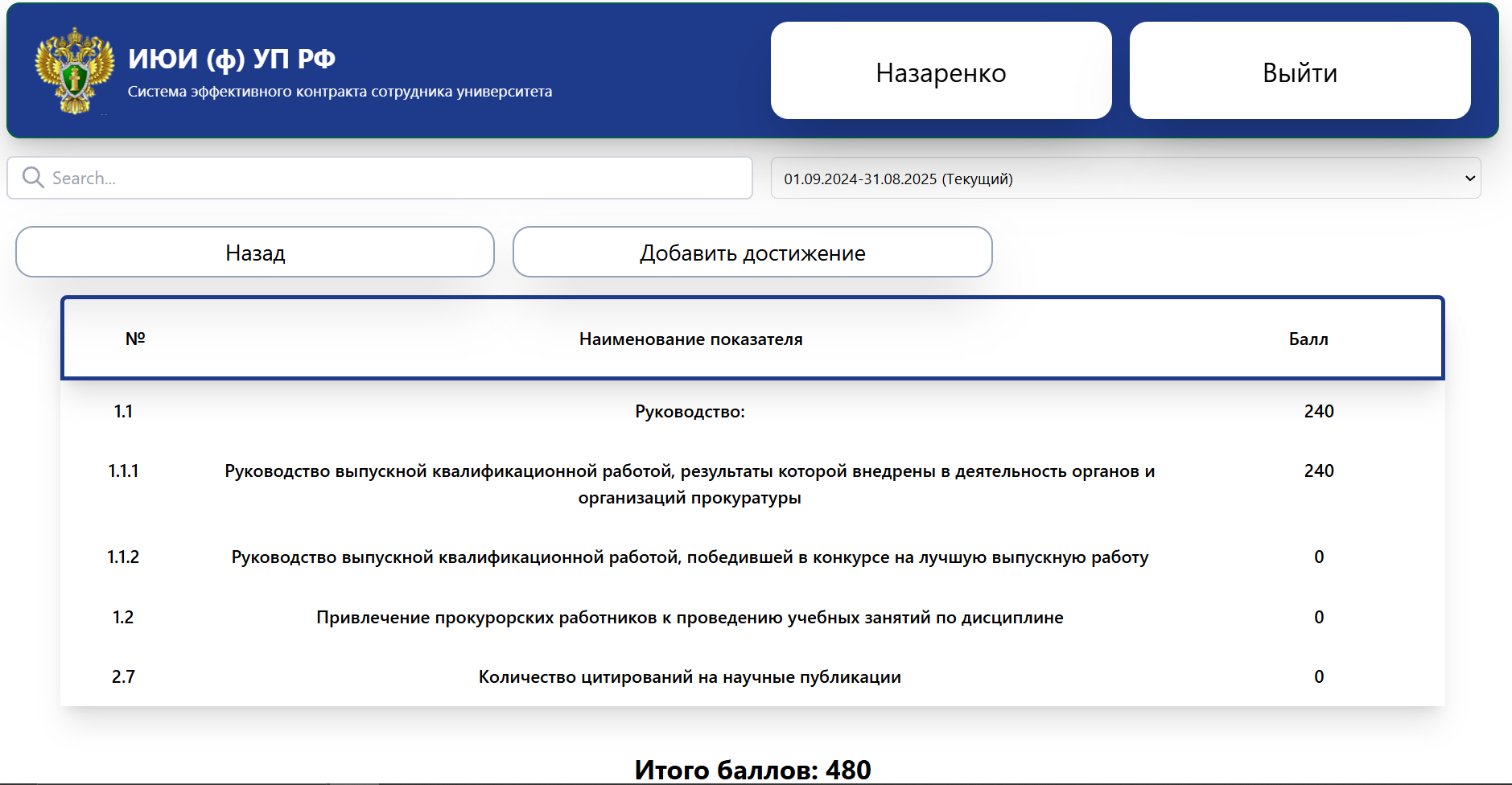


Рисунок 3.7.2 Личный кабинет пользователя

Для добавления нового достижения сотрудник нажимает на кнопку «Добавить достижение», появляется модальное окно добавления (рисунок 3.7.3). Сотрудник выбирает соответствующий показатель и подкатегорию, после чего заполняет необходимые данные, такие как наименование достижения, единица измерения и количество баллов. В зависимости от типа достижения, сотрудник может прикрепить подтверждающий документ в формате PDF, DOC, JPEG или PNG, либо указать ссылку на публикацию, если это требуется. После заполнения всех полей сотрудник подтверждает внесение данных, и достижение автоматически добавляется в его список.

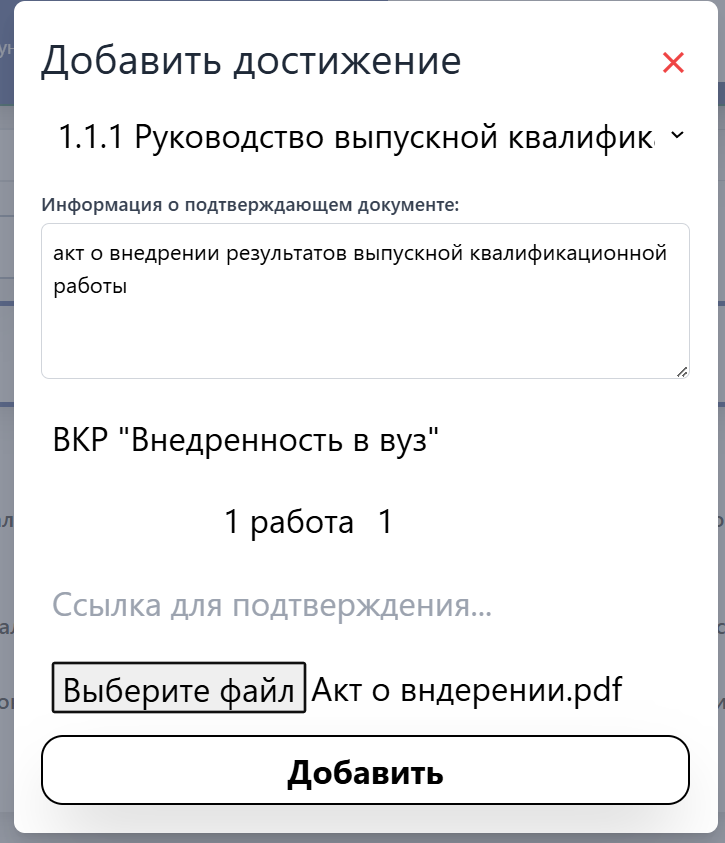


Рисунок 3.7.3 Добавление достижения

После успешного добавления достижения, сотрудник может просмотреть его в своем списке достижений (рисунок 3.7.4), где отображаются все внесенные данные, включая подтверждающие документы и ссылки. В случае необходимости, сотрудник может редактировать или удалить достижение, а также просмотреть детальную информацию о каждом из них.

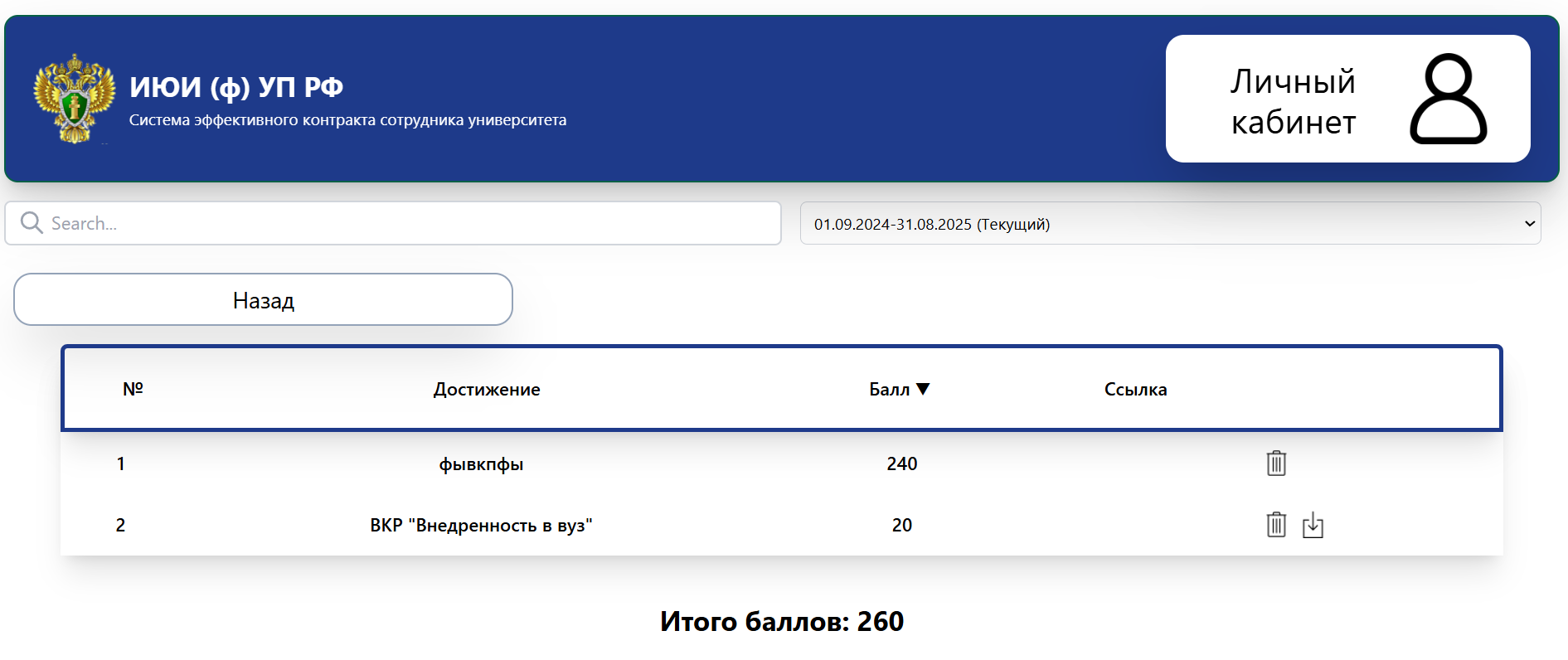


Рисунок 3.7.4 Проверка добавленного достижения

Таким образом, процесс работы с приложением для сотрудника является интуитивно понятным и удобным, что позволяет минимизировать временные затраты на внесение данных и повысить точность формирования рейтинга.

## 3.8 Возможное развитие системы

В настоящее время разработанная информационная система автоматизации формирования рейтинга сотрудников в рамках «Системы эффективного контракта» является важным инструментом, позволяющим существенно упростить процесс расчета KPI и формирования отчетности. Однако, как и любая современная система, она обладает потенциалом для дальнейшего совершенствования и расширения функциональных возможностей. Исходя из текущей ситуации, в которой большое количество предприятий любых направленность, в том числе учебных заведений, переходит на цифровые продукты, можно выделить несколько направлений, реализация которых позволит повысить удобство использования системы, интеграцию с другими информационными ресурсами и ее аналитические возможности.

Одним из ключевых направлений развития системы является интеграция с другими информационными системами университета, которые, с большой долей вероятности, будут внедряться в будущем. В перспективе возможно объединение с платформами управления учебным процессом, системами кадрового учета и финансового планирования и т.д. Это позволит автоматизировать обмен данными между системами, исключив необходимость ручного ввода информации. Например, данные о преподавательской нагрузке или участии сотрудников в научных мероприятиях могут поступать в систему эффективного контракта автоматически, что снизит вероятность ошибок и повысит оперативность обработки информации.

Не менее важным аспектом является расширение возможностей формирования отчетности. В настоящее время система поддерживает экспорт отчетов в формате Excel и PDF, однако дальнейшее развитие может включать разработку более сложных аналитических отчетов, таких как анализ динамики изменения KPI сотрудников за несколько лет, сравнительные отчеты между подразделениями и прогнозные модели. Такие возможности позволят руководству университета принимать более обоснованные управленческие решения на основе детализированной информации.

Существенное внимание должно быть уделено совершенствованию пользовательского интерфейса. Одним из приоритетов является внедрение адаптивного дизайна, что позволит сотрудникам и администраторам работать с системой не только с персональных компьютеров, но и с мобильных устройств, таких как планшеты и смартфоны. Это повысит удобство взаимодействия с системой, особенно в условиях необходимости оперативного доступа к информации. Улучшение интерфейса также может включать в себя модернизацию пользовательского опыта (UX/UI).

С учетом перспектив увеличения количества пользователей и объемов обрабатываемых данных, важной задачей является оптимизация производительности системы. Внедрение механизмов кэширования, оптимизация запросов к базе данных позволят повысить скорость обработки данных.

Развитие аналитических возможностей также является перспективным направлением модернизации системы. Внедрение модулей визуализации данных, таких как интерактивные графики, диаграммы и дашборды, позволит получить наглядное представление о рейтингах сотрудников и подразделений. Дополнительно можно предусмотреть возможность прогнозирования изменений рейтингов на основе исторических данных, что поможет руководству университета более эффективно планировать меры по повышению мотивации сотрудников.

Развитие системы уведомлений также представляется важной задачей. Автоматическое информирование сотрудников о сроках сдачи отчетов и необходимости внесения данных о достижениях позволит повысить уровень дисциплины и снизить риск пропуска значимых дат. Уведомления могут быть реализованы в виде сообщений на электронную почту или через интеграцию с мессенджерами, что обеспечит максимальную оперативность доставки информации. Кроме того, может быть предусмотрено информирование пользователей о важных изменениях в системе, таких как обновления функционала или корректировки в методологии расчета KPI.

Для повышения прозрачности работы системы и обеспечения удобства контроля может быть реализован механизм мониторинга активности пользователей. Это позволит администраторам отслеживать, кто и когда вносил изменения в систему, что будет способствовать повышению уровня ответственности сотрудников и предотвращению возможных ошибок. Также может быть предусмотрена автоматическая генерация отчетов по заданному расписанию, что позволит упростить процесс подготовки отчетности и снизить нагрузку на пользователей.

Таким образом, разработанная информационная система обладает значительным потенциалом для дальнейшего совершенствования. Реализация предложенных направлений развития позволит повысить ее функциональность, удобство использования и уровень интеграции с другими информационными ресурсами. Это, в свою очередь, будет способствовать дальнейшей цифровизации и автоматизации процессов в университете, повышая общую эффективность работы сотрудников и руководства.

## Заключение по разделу

В разделе, посвященном реализации проекта, были успешно выполнены все ключевые этапы разработки информационной системы автоматизации формирования рейтинга сотрудников. Была создана база данных, реализованы API-запросы для взаимодействия клиентской и серверной частей системы, а также разработан удобный и интуитивно понятный пользовательский интерфейс.

Клиентская часть системы, разработанная на основе фреймворка Vue.js, обеспечивает удобное взаимодействие пользователей с системой, включая просмотр рейтингов, внесение данных о достижениях и генерацию отчетов. Серверная часть, реализованная на Django, отвечает за обработку данных, расчет KPI и взаимодействие с базой данных PostgreSQL.

Текущая версия системы уже позволяет значительно упростить процесс формирования рейтинга сотрудников, сократить время на выполнение рутинных задач и минимизировать вероятность ошибок. Однако, как и любое программное обеспечение, система имеет потенциал для дальнейшего развития. В будущем возможно расширение функциональности системы, интеграция с другими информационными системами университета, а также улучшение пользовательского интерфейса и производительности.

Реализация проекта показала, что автоматизация процессов формирования рейтинга сотрудников является важным шагом в цифровизации университета. Внедрение системы позволит не только сократить временные и финансовые затраты, но и повысить общую эффективность работы сотрудников и руководства университета.

# 4 Экономическая часть

Для того, чтобы успешно вести весь цикл разработки программного обеспечения, в данном случае информационной системы эффективного контракта, необходимо, в первую очередь составить план всех работ на протяжении проектов. Наиболее универсальным инструментом в этом деле является диаграмма Ганта, при помощи которой можно смоделировать весь процесс разработки и наглядно увидеть все ее этапы. Для того, чтобы реализовать схему будет использоваться программный продукт MS Project, который предоставляет все необходимые инструменты, для составления плана работ, оценки ресурсов, подсчета общей стоимости проекта.

Всю основную работу ведет разработчик, лицо, к которому обратился университет, для решения проблем автоматизации процесса. На его плечи ложиться все, от анализа вводных, до финальных тестов при вводе в эксплуатацию.

Для полного понимания разработчиком особых моментов, которые могут возникать в процессе реализации проекта необходим консультант, в роли которого выступит работник IT-отдела университета, который будет консультировать разработчика по всем узким моментам в процессе, а также участвовать в итоговом вводе продукта в эксплуатацию. На рисунке 4.1 показан общий план выполнения работ проекта, а также лица и материалы, участвующие в процессе в качестве диаграммы Ганта. На рисунке 4.2 показан лист задач со всеми датами, назначенными для начала и завершения этапов. Общая длительность проекта, по текущим оценкам является 195 дней.

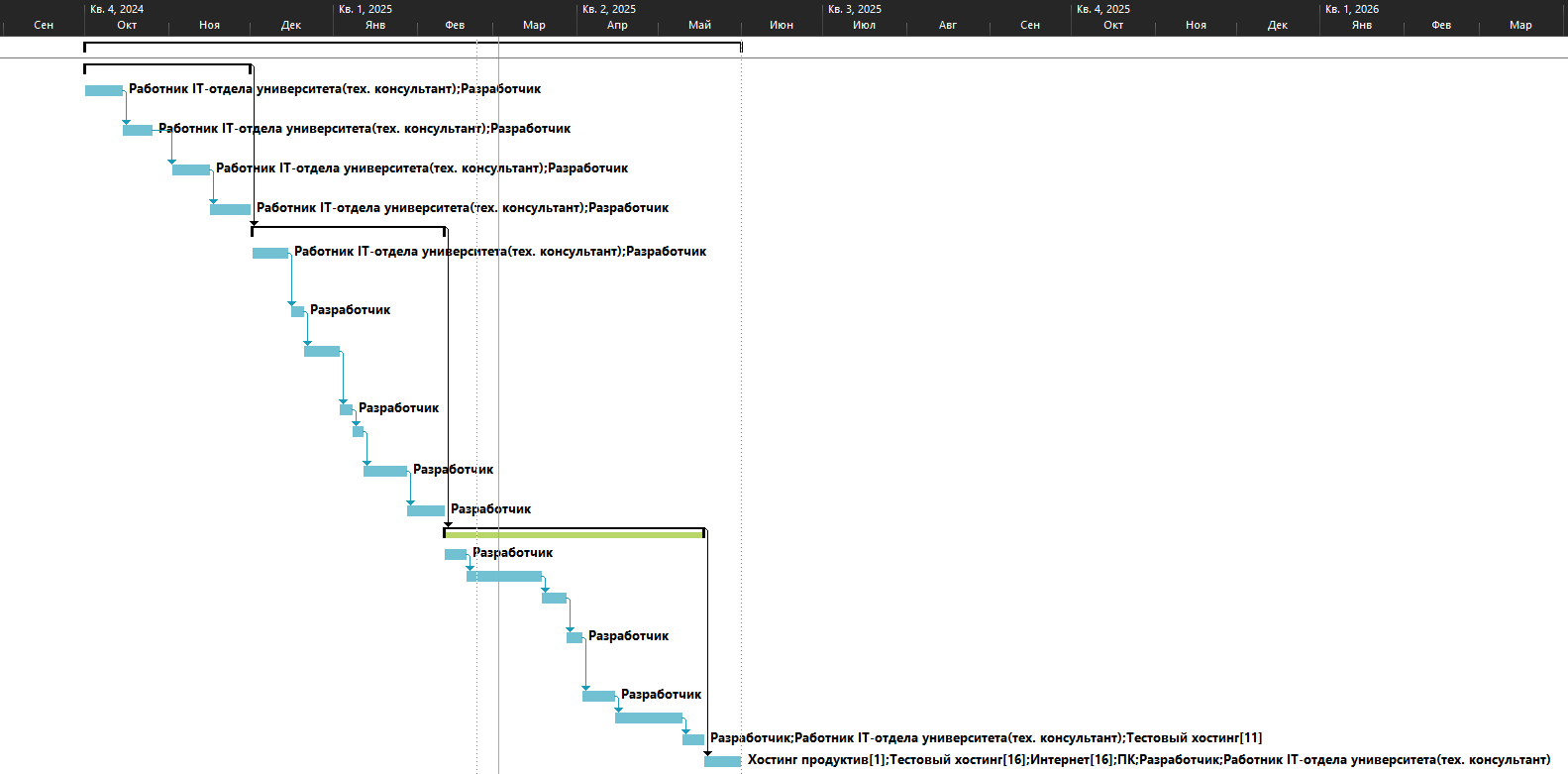


Рисунок 4.1 – Диаграмма Ганта проекта «Информационная система эффективного контракта ИЮИ (ф) УП РФ»

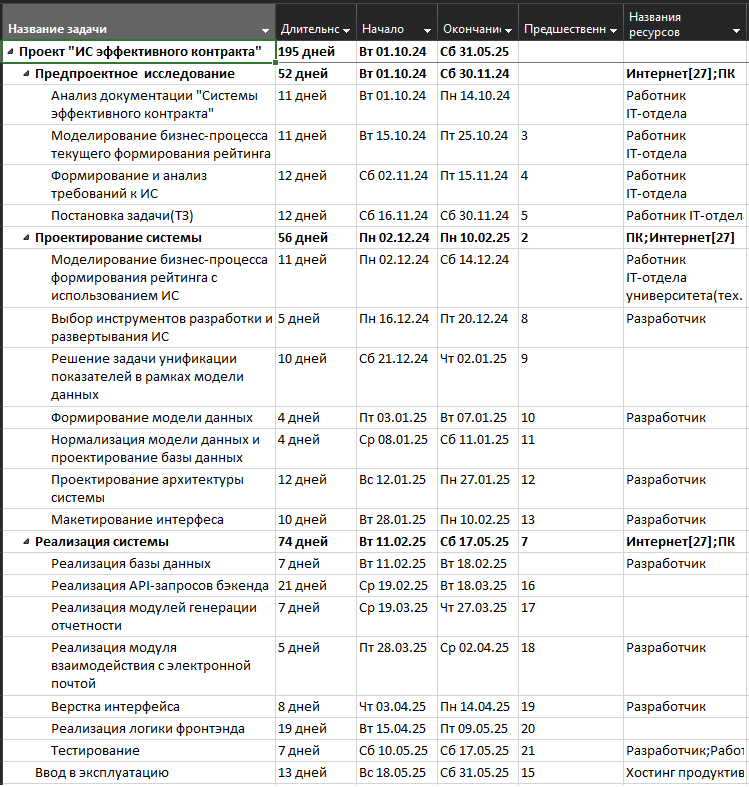


Рисунок 4.1 – Лист задач проекта «Информационная система эффективного контракта ИЮИ (ф) УП РФ»

Определение ресурсов также является неотъемлемой частью планирования проектов. Определение ставки лиц участвующих в проекте поможет определить общие затраты на человеко-часы в проекте, а введение ресурсов может узнать материальные затраты. В качестве ставки разработчик получает 100 руб./час, а консультант, в качестве надбавки к зарплате получает 100 руб./час за работу над новым проектом университета. В качестве ресурсов выступает персональный компьютер, который находиться в личном имении у разработчика, поэтому не несет за собой финансовых затрат, а также интернет, цена которого составляет 50 рублей в день. Кроме того, необходимо обеспечить тестовый и продуктовый хостинги. Тестовый хостинг арендуется по 10 руб./день, для проверки всей работоспособности проекта, а продуктовый хостинг, это выделенное место на серверах университета, который имеет достаточно свободных мощностей, чтобы обеспечить работоспособность ИС. На рисунке 4.3 представлен лист ресурсов, на котором отображены все ресурсы, которые используются в проекте

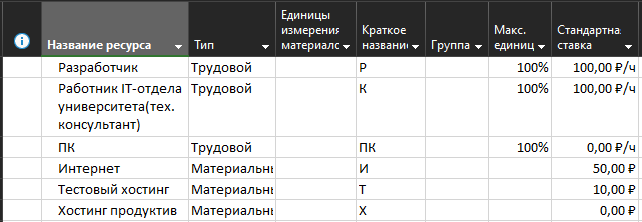


Рисунок 4.3 – Лист ресурсов проекта «Информационная система эффективного контракта ИЮИ (ф) УП РФ»

После всего моделирования задач и ресурсов проекта в MS Project открывается возможность получить отчетность по проекту для того, чтобы узнать наиболее важный момент всей разработки – «Общие затраты на проект».

При помощи графиков распределений затрат были сформированный отчеты, которые отражены на рисунке 4.4.



Рисунок 4.4 – Обзор затрат проекта «Информационная система эффективного контракта ИЮИ (ф) УП РФ»

Внедрение информационной автоматизированной системы эффективного контракта сотрудника университета является действительно важным этапом цифровизации и автоматизации работы во всем университете. В первую очередь, периоды формирования рейтинга, которые в данным момент перетягивают на себя 90% времени тех людей, которые за него ответственны, станут гораздо более простыми, за счет того, что основное время, которое тратилось на подсчет и перенос документов, будет нивелировано системой. Это влечет за собой не только разгрузку сотрудников, но и экономическую выгоду, поскольку теперь, то время, которое тратилось на этот процесс, будет уделено более важным вещам, а следовательно часть заработной платы, конкретно для этого процесса, будет уходить гораздо более эффективным путем.

Кроме того, система обеспечит централизованное хранение всех данных по рейтингам, что минимизирует риск потери данных, упразднит необходимость документооборота между этапами процесса, упростит поиск необходимых документов в любой период времени, будь то подача списка своих публикаций куда-либо, участие в конкурсе, в котором необходимо предоставить некоторые свои заслуги и т.д.

С точки зрения экономической выгоды, затраты на систему будут равняться сумме меньшей, чем месячная зарплата одного сотрудника, который будет ответственен за рейтинг, а в процессе их принимает участие несколько и, учитывая время того периода, который отводиться для его формирования, то выйдет, что затраты на разработку окупаться уже за первый год своего использования и внедрения. Также присутствуют факторы экономии бумаги при документообороте и незначительного количества времени у лиц, подписывающих и читающих эти документы в процессе, что в некотором роде, тоже экономит некоторую часть средств университета, выделенной для данного процесса.

Вложение средств в разработку системы автоматизации является более рациональным с экономической точки зрения, чем постоянные расходы на время ответственных за рейтинг. Автоматизация не только уменьшит объем рутинных задач, но и повысит общую эффективность работы, что в долгосрочной перспективе сделает процесс итоговой аттестации менее затратным и более продуктивным, позволит хранить всю информацию о деятельности преподавателей в одном месте, а также откроет путь к полной цифровизации рутинных процессов университета в будущем

# Заключение

В результате выполнения данной курсовой работы было успешно спроектировано и реализовано отказоустойчивое и дружественное к пользователю приложение для автоматизации формирования рейтинга сотрудников университета в рамках «Системы эффективного контракта», которое решает все поставленные заказчиком проблемы: сокращение времени, отводящееся на формирование рейтинга, разгрузка ответственных сотрудников, повышение качества и точности процесса, за счет компьютерного расчета и внутрисистемного оборота данных, вместо документооборота.

В процессе выполнения работы были выполнены следующие задачи:

* Анализ требований и задач, связанных с системой эффективного контракта;
* Формирование требований к системе;
* Применение методологии проектирования для создания проекта программного продукта;
* Выбор и использование соответствующих структур данных для эффективной реализации предметной области приложения;
* Разработка удобного пользовательского интерфейса, обеспечивающего комфортное взаимодействие с приложением для всех пользователей;
* Реализация MVP;
* Анализ экономической составляющей проекта.

Результатом данной работы стало создание надежного и удобного в использовании веб-приложения для просмотра и составления рейтинга сотрудников. Дальнейшее развитие приложения может включать расширение его функциональности и улучшение производительности.

Данное приложение, в некоторой степени, послужит отправной точкой цифровизации университета и перевода всех рутинных процессов на рельсы автоматизации. ИС может выполнять роль не только автоматизированного помощника в периоды формирования рейтинга, но и также, общевузовское хранилище всей информации, связанной с деятельностью преподавателя, что при дальнейшей цифровизации университета позволит встроить ее в экосистему продуктов, для более глубокого оборота информации в университете. В комбинации с другими, гипотетическими ИС в университете, возможно автоматическое заполнение данных из одной системы в другую, что влечет за собой еще большее упрощение процессов, потому внедрение данного продукта открывает для ИЮИ дверь в цифровое будущее.

Исходный код проекта находиться в открытом доступе на публичном репозитории GitHub [7].

# Список литературы

1. Структура // Иркутский юридический институт (филиал) федерального государственного казенного образовательного учреждения высшего образования «Университет прокуратуры Российской Федерации» URL: https://iagprf.org/struktura/ (дата обращения: 15.11.24).
2. Приказ "Об утверждении Положения об оценке эффективности и деятельности (балльной системе) научных и педагогических работников федерального государственного казенного образовательного учреждения высшего образования "Университет прокуратуры российской федерации"" от 24.07.2024 № 382 // Российская газета. - 24.07.2024
3. Оценка персональных результатов деятельности педагогических работников Университета прокуратуры Российской Федерации "Приложение №1 к Положению об оценке деятельности (балльной системе) научных и педагогических работников федерального государственного казенного образовательного учреждения высшего образования "Университет прокуратуры Российской Федерации"" от 24.07.2024 № 382 2024
4. Форма представления сведений (исходных данных) о результатах деятельности педагогических работников кафедры государственно-правовых дисциплин "Приложение №3 к Положению об оценке деятельности (балльной системе) научных и педагогических работников федерального государственного казенного образовательного учреждения высшего образования "Университет прокуратуры Российской Федерации"" от 24.07.2024 № 382 2024
5. Сведения о результатах деятельности "Приложение №4 к Положению об оценке деятельности (балльной системе) научных и педагогических работников федерального государственного казенного образовательного учреждения высшего образования "Университет прокуратуры Российской Федерации"" от 24.07.2024 № 382 2024
6. Эффективный контракт - автоматизация расчета стимулирующих выплат // клерк URL: https://www.klerk.ru/blogs/RAMZES/505843/ (дата обращения: 17.01.25).
7. IUI\_RatingProject // GitHub URL: https://github.com/YYEg/IUI\_RatingProject (дата обращения: 04.03.2025).
8. draw.io URL: https://app.diagrams.net/ (дата обращения: 10.12.24).
9. Ramus // github URL: https://github.com/Vitaliy-Yakovchuk/ramus/releases (дата обращения: 01.12.24).
10. Django documentation // The web framework Django URL: https://docs.djangoproject.com/en/5.1/ (дата обращения: (31.12.24)).
11. Vue guides // Vue.js URL: https://vuejs.org/guide/introduction.html (дата обращения: (11.01.25)).
12. Documentation // PostgreSQL URL: https://www.postgresql.org/docs/ (дата обращения: 15.11.24).
13. Notation // C4 Notation | Diagram URL: https://c4model.com/diagrams/notation (дата обращения: 24.02.2025).
14. Базы данных: Моделирование, проектирование, реализация // НОУ ИНТУИТ URL: https://intuit.ru/studies/courses/1001/297/lecture/7411 (дата обращения: 12.12.2024).
15. Моделирование данных: обзор // Habr URL: https://habr.com/ru/articles/556790/ (дата обращения: 13.12.2024).
16. Все что нужно для начала работы с Vue.js // Habr URL: https://habr.com/ru/companies/ruvds/articles/458324/ (дата обращения: 29.12.24).