

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГАОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных**  **технологий** | **Кафедра информационных технологий и вычислительных систем** |

**Романов Илья Олегович**Выпускная квалификационная работа  
по направлению подготовки   
09.04.04 «Программная инженерия»  
профиль «Технологии разработки интеллектуальных систем и программных комплексов»

**Исследование и разработка системы оценки качества проведённых занятий**

Регистрационный номер №\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Зав. кафедрой ИТиВС | \_\_\_\_\_\_\_\_\_\_\_\_\_ | к.т.н., доц. Новоселова О.В. |
| Научный руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_ | ст. преподаватель Бердюгин А.В. |
| Обучающийся | \_\_\_\_\_\_\_\_\_\_\_\_\_ | Романов И.О. |



**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных**  **технологий** | **Кафедра информационных технологий и вычислительных систем** |

|  |  |
| --- | --- |
|  | **«УТВЕРЖДАЮ»**  Заведующий кафедрой ИТиВС  к.т.н., доц. Новоселова О.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Задание**

на выполнение выпускной квалификационной работы   
по направлению подготовки 09.04.04 «Программная инженерия»  
профиль «Технологии разработки интеллектуальных систем и программных комплексов»

|  |  |
| --- | --- |
| Студент группы ИДМ-23-08 Романов И. О. | Научный руководитель  ст. преподаватель Бердюгин А.В. |

**Тема: «Исследование и разработка системы оценки качества проведённых занятий»**

1. **Описание задания на выполнение ВКР**
   1. **Тип ВКР –** исследовательскаяработа.
   2. **Цель исследования –** повышение эффективности оценивания занятия в университете за счёт разработки автоматизированной системы, обеспечивающей сбор статистических данных о качестве проводимых занятий.
   3. **Объект исследования –** управлениеобразовательными и научно-методическимипроцессами
   4. **Предмет исследования –** информационно-программные средства для поддержки образовательных процессов.
   5. **Методы исследования –** системный анализ, методы обработки данных, практическое моделирование, разработка автоматизированных систем.
   6. **Задачи исследования:**
      1. Изучить и проанализировать современные языки программирования и фреймворки в области разработки ПО, современные подходы к разработке серверных приложений.
      2. Разработать модель разрабатываемой автоматизированной системы.
      3. Выбрать и обосновать средства реализации.
      4. Разработать автоматизированную систему контроля качества проведённых занятий с предоставлением целевых метрик.
2. **Требования к выполнению ВКР**
   1. **Соблюдение требований законодательной базы и стандартов**
      1. Образовательная программа высшего образования в магистратуре ФГБОУ ВО «МГТУ «СТАНКИН» по направлению подготовки 09.04.04 «Программная инженерия» для направленности (профиля) подготовки «Технологии разработки интеллектуальных систем и программных комплексов» (утв. 13.04.2022).
      2. Внутренний нормативный документ. П 01-04/264/2017. Положение о государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры: [утверждено приказом ректора от 31.08.2017 г. №431/1, одобрено решением ученого совета Университета от 31.08.2017 г., протокол № 10/17] – ФГБОУ ВО «МГТУ «СТАНКИН».
      3. Внутренний нормативный документ. П 01-04/438/2021. Положение о выпускной квалификационной работе обучающихся по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры: [утверждено приказом врио ректора от 30.03.2021 г. №177/1, одобрено решением ученого совета Университета от 25.12.2020 г., протокол № 11/20] – ФГБОУ ВО «МГТУ «СТАНКИН».
   2. **Дополнительные требования**
      1. Оформление раздела «Список литературы» по национальному стандарту ГОСТ Р 7.0.100-2018 “Библиографическая запись. Библиографическое описание. Общие требования и правила составления”.
      2. Результаты исследования должны быть опубликованы в виде научных статей и тезисов докладов (не менее 4), а также доложены на научно-технических конференциях и семинарах.
   3. **Срок сдачи оформленной квалификационной работы на кафедру – вторая декада мая 2025 г.**

|  |  |  |
| --- | --- | --- |
| Исполнитель |  | Романов И. О. |
| Научный руководитель,  старший преподаватель |  | Бердюгин А.В. |

**АВТОРЕФЕРАТ**

Выпускной квалификационной работы

Студента группы ИДМ-23-08 ФГБОУ ВО «МГТУ «СТАНКИН»

**Романов Илья Олеговч**

по направлению

09.04.04 «Программная инженерия»

на тему

**«Исследование и разработка системы оценки качества проведённых занятий»**

**Актуальность работы.**

**Цель**

Для достижения поставленной цели в работе были поставлены и решены следующие **задачи**:

* + 1. Провести **Научная новизна** состоит в создании единой коммуникационной онлайн-системы для преподавателей кафедры, замещающей собой прежние разрозненные каналы коммуникаций.

**Практическая ценность работы** состоит в

**Структура работы.** ВКР состоит из Введения, пяти глав глав, Заключения, Списка литературы (12 наименований) и Приложения (стр.). Выпускная квалификационная работа содержит 95 страниц текста, рисунков – 57, таблиц – 19, использованных источников – 12.

**Апробация работы.**

**Выводы по работе:**

1. В ходе

**Список работ, опубликованных по теме ВКР:**

1. «

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 7](#_Toc13169)

[ГЛАВА 1. РАЗРАБОТКА ТРЕБОВАНИЙ](#_Toc31841)

[К ПРОГРАММНОМУ ПРОДУКТУ 8](#_Toc31841)

[1.1. ПОНЯТИЙНЫЙ АППАРАТ 8](#_Toc7641)

[1.1.1. Качество проведённых занятий 8](#_Toc3717)

[1.1.2. Проведённое занятие 8](#_Toc8057)

[1.1.3. Целевые метрики 8](#_Toc23845)

[1.2. ОБЩИЕ СВЕДЕНИЯ 9](#_Toc17973)

[1.2.1. Полное наименование системы и её условное обозначение 9](#_Toc2488)

[1.2.2. Назначение и цели создания системы 9](#_Toc21577)

[1.2.3. Цели создания системы 9](#_Toc16828)

[1.3. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ 9](#_Toc11423)

[1.3.1. WebAsk.io 10](#_Toc16948)

[1.3.2. Электронная образовательная среда (СЭО 3KL Русский Moodle) 11](#_Toc28393)

[1.3.3. Stepik 13](#_Toc4058)

[1.4. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ 13](#_Toc10630)

[1.5. ТРЕБОВАНИЯ К СИСТЕМЕ 14](#_Toc20984)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ 16](#_Toc30575)

[2.1. ДИАГРАММА ПРЕЦЕДЕНТОВ 16](#_Toc27921)

[2.2. СХЕМА ДАННЫХ 18](#_Toc21194)

[2.2.1. Процесс «Расписание» 18](#_Toc9144)

[2.2.2. Процесс «Аутентификация и авторизация» 21](#_Toc26832)

[2.2.3. Процесс «Тестирование» 23](#_Toc30543)

[2.2.4. Процесс «Оценка качества проведённого занятия» 24](#_Toc25131)

[2.3. ДАННЫЕ СТУДЕНТА. МОДУЛЬНЫЙ ЖУРНАЛ 25](#_Toc3967)

[ГЛАВА 3. РЕАЛИЗАЦИЯ 26](#_Toc27382)

[3.1. ОПИСАНИЕ ТРЕБОВАНИЙ К СРЕДСТВАМ РЕАЛИЗАЦИИ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ 26](#_Toc26280)

[3.1.1. Пользовательский интерфейс 26](#_Toc24578)

[3.1.2. Хранение данных 27](#_Toc10905)

[3.1.3. Серверное приложение 28](#_Toc29459)

[3.2. РЕАЛИЗАЦИЯ 30](#_Toc31829)

[3.2.1. Реализация пользовательского интерфейса 31](#_Toc26159)

[3.2.1.1. Расписание занятий 32](#_Toc20144)

[3.2.1.2. Тестирование и опросы 34](#_Toc15924)

[3.2.1.3. Модульный журнал. OAUTH 2.0 провайдер 39](#_Toc25341)

[3.2.2. Реализация серверной части 42](#_Toc3922)

ВВЕДЕНИЕ

Темой выпускной квалификационной работы является «Исследование и разработка системы оценки качества проведённых занятий». Данная тема подразумевает под собой проектирование и реализацию программного продукта, полностью удовлетворяющего требованиям конечного потребителя, а именно – учреждение высшего образования (здесь и далее – Общество). Цель данного программного продукта – автоматизация процесса оценки качества проведённых занятий с предоставлением целевых метрик.

Первоначально следует исследовать рынок программного обеспечения на наличие уже готовых программных решений, с помощью которых может быть удовлетворена потребность Общества в программном продукте.

ГЛАВА 1. РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

* 1. **ПОНЯТИЙНЫЙ АППАРАТ**

Перед началом разработки требований к программному продукту, необходимо определить понятийный аппарат в выбранной предметной области.

* + 1. Качество проведённых занятий

Под качеством проведённых занятий, в данной предметной области, подразумевается объективная оценка, основанная на численных методах. В данном случае оценка может быть двунаправленная – как роль «студент» может оценить качество проведённого занятия, так и роль «преподаватель» может выставить свою оценку для конкретной группы или потока.

* + 1. Проведённое занятие

Под проведённым занятием, в данной предметной области, подразумевается событие, обозначенное в учебном плане, связанное с Обществом, в котором участвуют роли «Студент» и «Преподаватель».

* + 1. Целевые метрики

  Под целевыми метриками, в данной предметной области, подразумевается набор численных агрегатов или словарно-семантических диаграмм, собранных в ходе выполнения целевых бизнес-процессов, наглядно иллюстрирующих объективную, с точки зрения статистики, но субъективную с точки зрения методологии, оценку качества проведённых занятий.

ОБЩИЕ СВЕДЕНИЯ

* + 1. Полное наименование системы и её условное обозначение

Полное наименование – «Система оценки качества проведённых занятий». Шифр – «СОКПЗ»

* + 1. Назначение и цели создания системы

Разрабатываемая система предназначается для оценки качества проведённых занятий. СОКПЗ должна создаваться в качестве единой точки сбора, агрегации и предоставления агрегированных данных в виде аналитических отчётов оценок качества проведённых занятий.

СОКПЗ служит для обеспечения:

1. Предоставления данных о качестве проведенных занятий Общества.
2. Хранение и агрегация данных в рамках системы.
3. Единую точку входа для как обучающихся общества, имеющих роль «Студент», так и для роли «Преподаватель», для возможности выставления оценки качества проведения занятий.
   * 1. Цели создания системы

В качестве основных целей создания системы можно выделить:

1. Формирование статистических данных о качестве проведения занятий.
2. Агрегация и расчёт статистических данных по институтам, кафедрам и иным административным единицам общества, для дальнейшего прикладного применения полученных сведений.

СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

Чтобы сформулировать требования для актуальной системы оценки качества проведённых занятий со сбором, агрегацией и анализом целевых метрик, необходимо провести анализ существующих решений, для того, чтобы при разработке функциональных и нефункциональных требований к системе учитывать все актуальные тенденции на рынке.

Далее будут рассмотрены самые распространённые, по общей функциональности, системы оценок.

* + 1. WebAsk.io

WebAsk.io – это российский сервис проведения онлайн-тестирований, анкетирования, маркетинговых и продуктовых исследований.

WebAsk решает следующие задачи:

1. Изучение клиентского опыта– анализ конкурентов, изучение потребностей покупателей, сбор обратной связи, оценка уровня лояльности.
2. Анализ продукта **–** оценка логотипа, макетов, дизайна упаковок, UX-тестирование, сравнение с конкурентами.
3. Проверка гипотез **–** тестирование концептов, изучение спроса.
4. Маркетинговые исследования - исследование рынка, проверка креативов, оценка эффективности рекламных кампаний, изучение ЦА, измерение уровня знания бренда.
5. Работа с персоналом - оценка удовлетворенности сотрудников, аттестации персонала, сбор обратной связи, тестирование персонала, опросы сотрудников.
6. Образовательные учреждения - психологические тестирования, исследование целей и ценностей обучающихся, тестирование учащихся, оценка качества занятий.

WebAsk.io предоставляет инструментарий для создания открытых и закрытых опросов. По проведению опроса, реализован функционал просмотра итоговой статистики. Данное программное обеспечение является платным, и предоставляется по подписке. В бесплатном тарифе есть ряд существенных ограничений – доступно только 3 опроса на аккаунт, 100 ответов в месяц и 10 элементов на опрос.

* + 1. Электронная образовательная среда (СЭО 3KL Русский Moodle)

В качестве примера электронной образовательной среды автор предлагает рассмотреть электронную образовательную среду ФГБОУ ВО МГТУ «СТАНКИН». Данная электронная образовательная среда реализована на базе СЭО (среда электронного обучения) 3KL (Русский Moodle). 3KL (Русский Moodle) – среда электронного образования, которая предоставляет инструменты для создания учебных курсов и организации учебного процесса.

Данная среда адаптирована под нужды российских заказчиков и применяется в корпоративном секторе, образовании, сфере дополнительного образования.

СДО 3KL Русский Moodle является профессиональной версией базовой системы Moodle и имеет с ней полную совместимость, распространяется компанией «Открытые Технологии».

В отличие от базовой версии, СЭО 3KL Русский Moodle поставляется с техподдержкой и содержит более 50-ти дополнительных модулей, разработанных компанией «Открытые Технологии»

Профессиональная версия СЭО 3KL® Русский Moodle включает в себя следующий уникальный функционал:

1. Панель управления СЭО 3КL - инструмент администрирования, предназначенный для работы с основными объектами системы: курсами, пользователями, глобальными группами. Позволяет выполнять массовые действия с объектами системы, например, записывать/отчислять глобальные группы пользователей на выбранные курсы. Предоставляет широкие возможности для анализа (пользовательской активности, использования и востребованности курсов и т.п.) при помощи удобного интерфейса редактируемых вкладок, содержащих настраиваемые таблицы параметров объектов СЭО 3КL. Создаваемые вкладки и их настройки индивидуальны для каждого пользователя. Предусматривает выгрузку данных из вкладок в файл с возможностью выбора формата сохранения, для последующего формирования статистической и аналитической отчетности.
2. Модуль «Электронный деканат» - управляющая надстройка системы для управления учебными процессами на основе учебных планов, параллелей и академических групп. Позволяет распределить объекты учебного процесса по иерархической структуре подразделений с разграничением прав доступа. Включает необходимую электронную документацию и отчетность, автоматизацию взаимодействия участников и индивидуальные траектории, значительно экономит время регламентных процедур, сохраняет историю всех процессов и событий.
3. Витрина курсов - графический каталог с широчайшими возможностями настройки оформления как списка (плиток) курсов и категорий, так и описательной страницы отдельно взятой дисциплины (курса). Предназначена для привлечения слушателей и онлайн-продаж курсов. Может использоваться для организации массовых онлайн-курсов. Каталог содержит дерево категорий курсов, списки курсов, страницу публичной информации о курсе, вывод мета-информации, плитки и иконки, встроенный поиск и фильтрацию курсов по названиям, датам, ценам, преподавателям и любым настраиваемым полям. Витрина может отображаться на любых страницах портала в любом удобном пользователю регионе интерфейса СЭО 3KL.
4. Система учета достижений и целей «Портфолио» позволяет организовать индивидуальный план развития персонала, формировать личные списки достижений пользователя, осуществлять целевое планирование в обучении и использоваться как инструмент для формирования кадрового резерва. Учет достижений включает подсистемы расчета рейтинга, модерации и фильтрации. Соответствует стандарту ФГОС 3++.

Данная платформа предлагает обширный функционал для ведения образовательной деятельности в цифровом формате.

Для эксплуатации данного программного обеспечения необходимо приобрести лицензию. В стоимость лицензии входит установка, техническая поддержка и обновление продукта, а также доступ к базе знаний и обучающим вебинарам.

* + 1. Stepik

**Stepik -** российская образовательная платформа и конструктор бесплатных и платных открытых онлайн-курсов и уроков. Данная платформа предоставляет интерфейс, который позволяет любому зарегистрированному создавать интерактивные обучающие онлайн-курсы, используя аудио, видео, текст и разнообразные задачи с автоматической проверкой и моментальной обратной связью.

В процессе обучения пользователи могут вести обсуждения между собой и задавать вопросы преподавателю на форуме. Основные охватываемые курсами дисциплины — программирование, математика, биоинформатика и биология, экономика; основной язык курсов — русский, есть курсы на английском языке. По состоянию на 2020 год на платформе зарегистрировано 5 миллионов пользователей.

АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

На основе сведений о существующих решениях, приведена таблица 1.1 представляющая из себя сравнительную характеристику функциональных свойств этих решений.

Таблица 1.1 Сравнительная характеристика существующих решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Анализ вовлеченности студентов | Оценка проведённого занятия ролью «Преподаватель» | Форма для создания тестовых заданий | Агрегация метрик по различным административным единицам |
| Google forms | - | - | + | - |
| СЭО 3KL  Русский Moodle | - | - | + | - |
| Stepik | - | - | + | - |

Пояснения по характеристикам в сводной таблице:

* Оценка проведённого занятия - прошедшие занятие участники с ролью «студент» могут оставить свою оценку/отзыв о качестве проведённого занятия;
* оценка группы - участник процесса с ролью «Преподаватель» может оценить конкретное множество участников с ролью «студент» под названием «группа»/«поток», по набору конкретных характеристик, например: количество, вовлеченность и результат «завершающего» краткого тестирования;
* форма для создания тестовых заданий - характеристика, описывающая наличие удобной пользовательской формы для создания «завершающего» тестового задания, с целью прохождения этого задания студентами;
* агрегация метрик по различным административным единицам - характеристика, определяющая возможность агрегации метрик по административным единицам. В предметной области «Образование» такими единицами могут стать: предмет, факультет, кафедра, институт, поток, группа, подгруппа и т.д;

ТРЕБОВАНИЯ К СИСТЕМЕ

На основе сравнительной характеристики существующих решений, сформулированные следующие функциональные требования к системе:

1. Система должна предоставлять единую точку входа для пользователей с различными ролями.
2. Система должна обеспечить способ получения, хранения, агрегации, отображения, и формирования отчётов по итогам оценивания качества проведённых занятий по различным административным единицам.
3. Процессы авторизации и аутентификации пользователей системы должны быть реализованы средствами Системы.
4. Технологические операции, выполняемые на технических средствах Системы должны быть полностью автоматизированы.
5. Результатом формирования отчётов системы должны быть заранее согласованного с заказчиком формата.
6. В системы должны быть включены роли студента, преподавателя и аудитора.
7. Роль «Аудитор» включает в себя роли «Студент» и «Преподаватель».

ГЛАВА 2. ПРОЕКТИРОВАНИЕ

В данной главе описаны независимые от средств реализации аспекты проектирование программной системы - диаграмма прецедентов, схемы данных, внешние интеграции и т.п.

ДИАГРАММА ПРЕЦЕДЕНТОВ

На рисунке 2.1 приведена диаграмма прецедентов информационной системы для оценки качества проведённых занятий. Данная диаграмма отражает взаимодействие между участниками процесса и функциональными возможностями системы.

Основными акторами системы являются преподаватель, студент и внешний аудит. Преподаватель выполняет такие действия, как составление критериев оценки занятий и тестов, анализ статистики данных, просмотр обратной связи от обучающихся, а также генерацию отчётов. Обучающийся взаимодействует с системой путём выполнения тестов, предоставления письменной обратной связи и выставления оценок за занятия. Аудитор выступает в роли внешнего участника, взаимодействующего с системой, например, для получения отчётов или анализа данных.

Ключевые прецеденты включают авторизацию, которая является отправной точкой для работы с системой. Преподаватель может составлять критерии оценки занятий и тесты, что расширяет базовый процесс авторизации. Обучающиеся имеют возможность просматривать критерии, оставлять обратную связь и проходить тестирование. Система также поддерживает генерацию отчётов на основе собранных данных, анализ статистики и анонимных выборок для оценки эффективности занятий.

Связи между прецедентами обозначены с помощью зависимостей <<include>> и <<extend>>. Например, анализ эффективности занятий обязательно включает ознакомление с критериями, а составление критериев оценки расширяет процесс авторизации.

Диаграмма наглядно демонстрирует структуру взаимодействия пользователей с системой, обеспечивающей оценку качества учебных занятий, сбор обратной связи и анализ результатов.

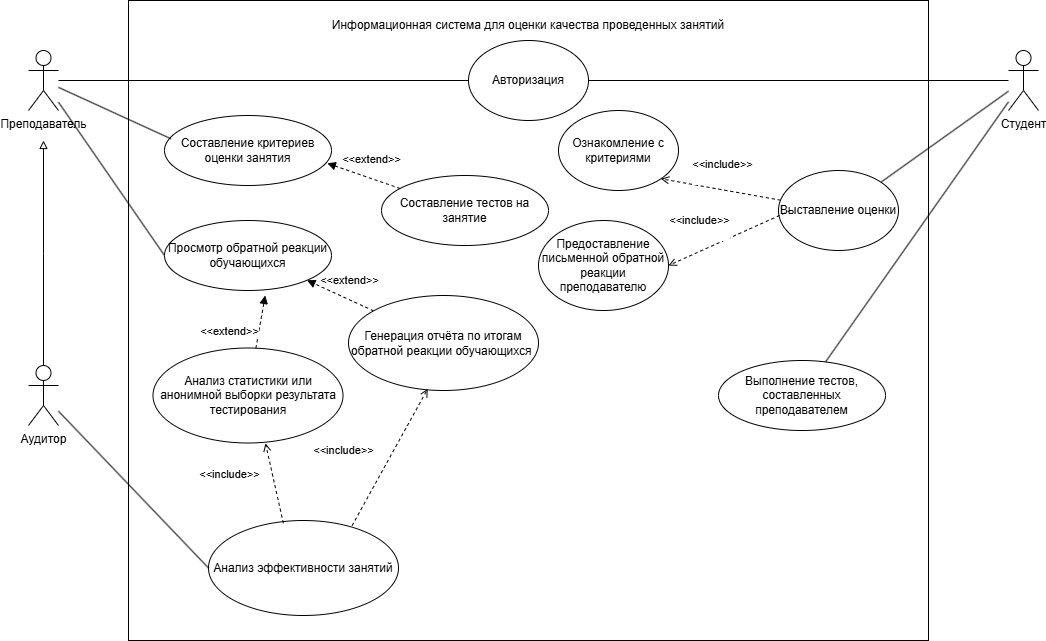


Рис. 2 Диаграмма прецедентов СОКПЗ

СХЕМА ДАННЫХ

На рисунке 2.1 представлена полная схема данных СОКПЗ.

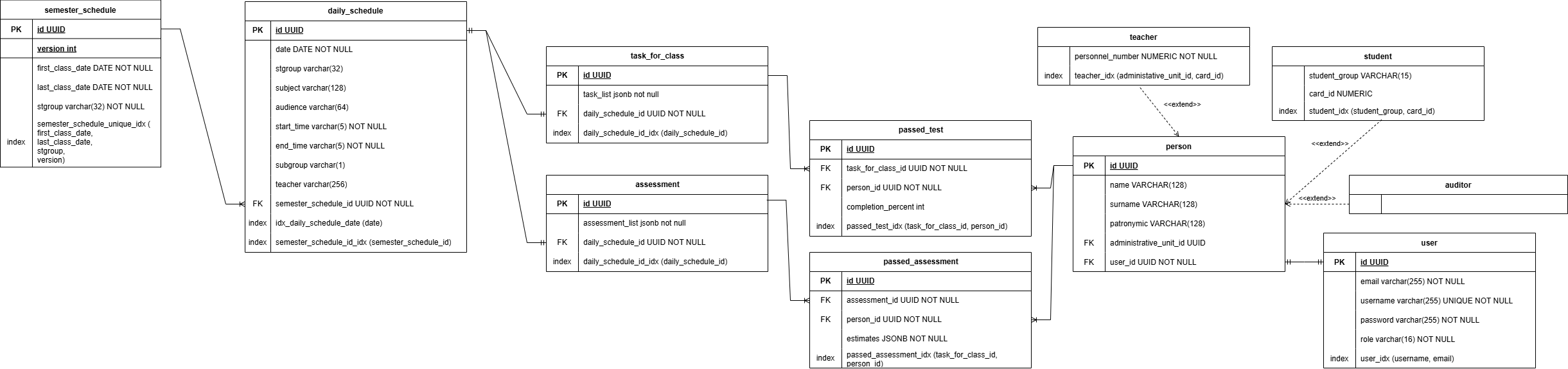


Рис. 2.1 Схема данных для процесса СОКПЗ

В следующих пунктах текущего раздела будут детально описаны процессы, представленные на схеме данных.

* + 1. Процесс «Расписание»

Схема базы данных на рисунке 2.2 для процесса создания расписания включает четыре основные таблицы, связанные между собой.

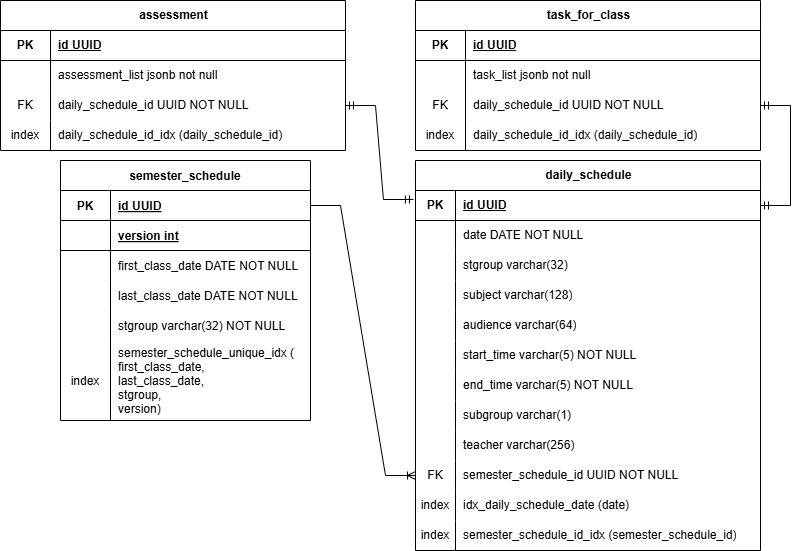


Рис. 2.2 Схема данных для процесса «Расписание»

Процесс "Расписание" в системе оценки качества учебных занятий представляет собой многоуровневую структуру, обеспечивающую планирование, проведение и анализ эффективности образовательного процесса. В его основе лежит семестровое расписание, которое определяет временные рамки учебного периода для конкретной группы студентов, включая даты начала и окончания занятий. Это общее расписание детализируется через ежедневные планы занятий, содержащие информацию о времени проведения, аудитории, преподавателе, типе занятия и учебном предмете.

Каждое занятие может быть связано с набором учебных заданий, представленных в структурированном формате, что позволяет адаптировать требования под конкретный урок. Для оценки качества проведённых занятий используется система критериев, также хранящаяся в гибком формате, что даёт возможность настраивать параметры оценивания в зависимости от дисциплины или методики преподавания.

Сбор данных о результатах обучения осуществляется через фиксацию выполнения заданий студентами, включая процент освоения материала, а также через качественные оценки по установленным критериям. Это позволяет анализировать эффективность занятий как с точки зрения усвоения знаний, так и с позиции организации учебного процесса. Взаимосвязь между семестровым планированием, ежедневными занятиями, учебными заданиями и системой оценки создаёт целостный механизм для управления образовательным процессом и постоянного повышения его качества.

Таблица semester\_schedule хранит информацию о семестровом расписании. Она содержит уникальный идентификатор (id UUID), версию расписания (version int), даты первого и последнего занятий (first\_class\_date и last\_class\_date типа DATE с ограничением NOT NULL), а также идентификатор группы (stgroup varchar(32) NOT NULL). Для обеспечения целостности данных в таблице определен уникальный индекс semester\_schedule\_unique\_idx, включающий даты занятий, идентификатор группы и версию расписания.

Таблица daily\_schedule содержит детализированную информацию о каждом занятии. Первичный ключ таблицы - id UUID. Для связи с семестровым расписанием используется поле semester\_schedule\_id UUID NOT NULL. В таблице хранятся такие данные как дата занятия (date DATE NOT NULL), идентификатор группы (signoup varchar(32)), название предмета (subject varchar(128)), аудитория (audience varchar(64)), время начала и окончания (start\_time и end\_time varchar(5) NOT NULL), подгруппа (subgroup varchar(1)) и преподаватель (teacher varchar(256)). Для оптимизации запросов созданы индексы: idx\_daily\_schedule\_date по дате занятия и semester\_schedule\_id\_idx по идентификатору семестрового расписания

Таблица task\_for\_class предназначена для хранения тестовых заданий к занятиям. Первичный ключ - id UUID. Связь с ежедневным расписанием осуществляется через поле daily\_schedule\_id UUID NOT NULL с соответствующим индексом daily\_schedule\_id\_idx. Список тестовых заданий хранится в формате JSON в поле task\_list jsonb NOT NULL.

Таблица assessment является ключевым элементом системы мониторинга качества учебного процесса, обеспечивающим хранение и обработку критериев оценки проведенных занятий. В её структуре используется уникальный идентификатор id типа UUID, выполняющий роль первичного ключа и гарантирующий однозначную идентификацию каждой записи. Основное содержание таблицы представлено полем assessment\_list формата jsonb, которое содержит гибко настраиваемый набор оценочных критериев, включая параметры методической эффективности, уровень вовлеченности студентов и соответствие образовательным стандартам.

Связи между таблицами организованы следующим образом: таблица daily\_schedule связана с semester\_schedule через поле semester\_schedule\_id, а task\_for\_class и assessment связана с daily\_schedule через daily\_schedule\_id. Такая структура позволяет эффективно организовать данные о расписании занятий и связанных с ними тестовых заданиях, обеспечивая целостность данных и быстрый доступ к необходимой информации.

* + 1. Процесс «Аутентификация и авторизация»

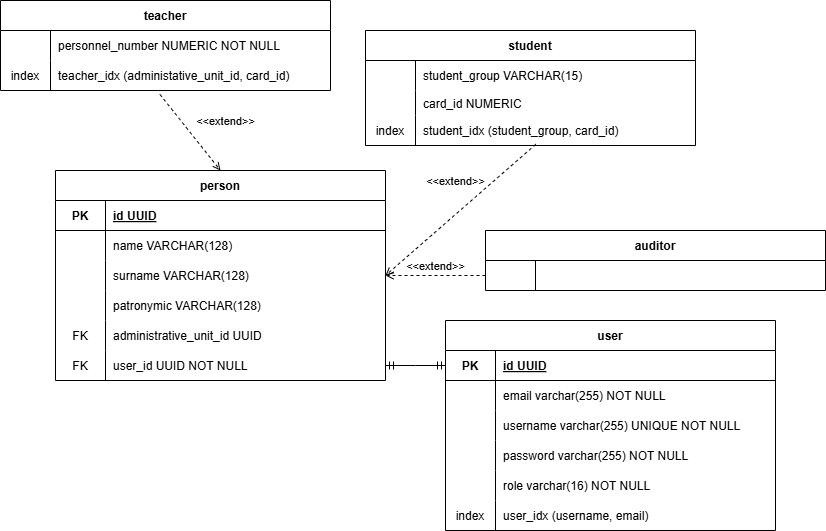


Рис. 2.3 Схема данных процесса «Авторизация и аутентификация»

Схема базы данных для платформы оценки качества учебных занятий, изображённая на рисунке 2.3, построена на основе наследования сущностей, где таблицы auditor (аудитор) и student (студент) и teacher (преподаватель), расширяют базовую лоигку таблицы person (базовый участинк учебного процесса) используя одинаковый набор атрибутов. Центральным элементом системы является таблица user, содержащая общие учетные данные для всех типов пользователей.

Таблица user включает обязательные поля для авторизации: уникальный идентификатор (id UUID) как первичный ключ, электронную почту (email varchar(255) с ограничением NOT NULL, уникальное имя пользователя (username varchar(255) с ограничениями UNIQUE и NOT NULL, зашифрованный пароль (password varchar(255) NOT NULL, а также роль пользователя (role varchar(16)) NOT NULL, которая определяет тип учетной записи (преподаватель, студент или аудитор). Для ускорения поиска создан составной индекс user\_idx по полям username и email.

Таблицы teacher, auditor и student имеют идентичную структуру, что реализует паттерн наследования в реляционной модели. Каждая из этих таблиц содержит: первичный ключ (id UUID), персональные данные (name, surname и patronymic типа VARCHAR(128)), служебный идентификатор (personal\_id NUMERIC), ссылку на структурное подразделение (administrative\_unit UUID NOT NULL) и обязательную связь с учетной записью через поле user\_id UUID NOT NULL. В таблице student дополнительно присутствуют поля для работы с учебными группами (student\_group VARCHAR(15)) и системами идентификации (card\_id).

Связи между таблицами организованы следующим образом: каждая из таблиц-наследников (teacher, auditor, student) связана с основной таблицей user через поле user\_id, которое является внешним ключом, ссылающимся на id в таблице user. Такая архитектура позволяет:

* Обеспечить единый механизм авторизации для всех типов пользователей
* Сохранять специализированные данные для каждой роли
* Поддерживать целостность данных через внешние ключи
* Реализовать разграничение прав доступа на основе ролей

Для оптимизации запросов в таблицах-наследниках созданы соответствующие индексы, включая student\_idx для работы со студенческими группами.

* + 1. Процесс «Тестирование»

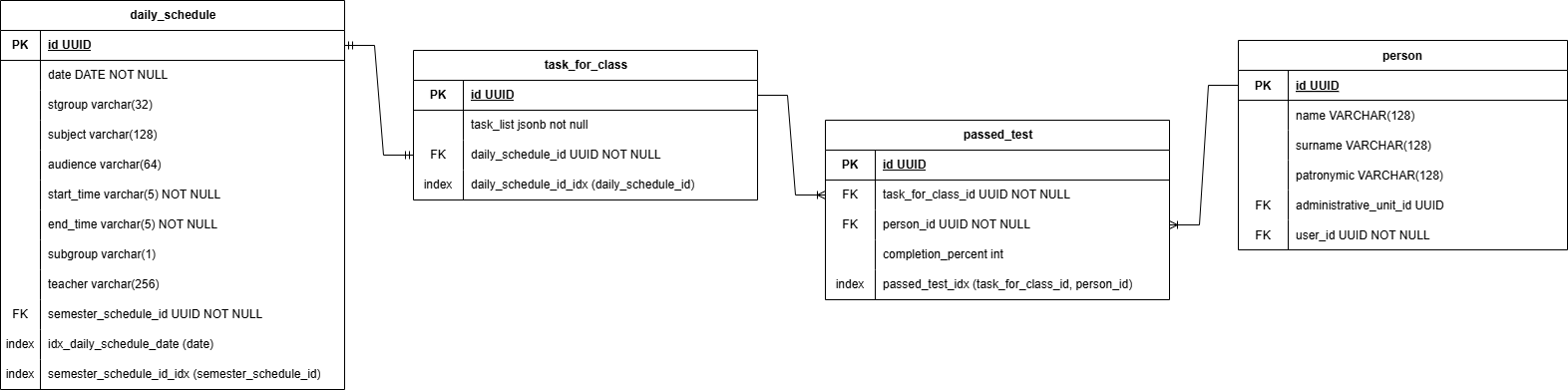


Рис. 2.4 Схема данных процесса «Тестирование»

Процесс тестирования представляет собой многоуровневую систему оценки знаний студентов, интегрированную в общую структуру учебного расписания. Основой процесса выступает таблица daily\_schedule, содержащая информацию о запланированных учебных занятиях с указанием даты, времени, аудитории, преподавателя и других параметров проведения урока. Каждое занятие может быть связано с набором учебных заданий через таблицу task\_for\_class, где в структурированном json-формате хранится перечень задач и тестовых материалов, подготовленных для студентов.

Результаты прохождения тестов фиксируются в таблице passed\_test, которая связывает конкретного студента из таблицы person с набором заданий. Поле completion\_percent количественно отражает степень выполнения тестовых заданий, позволяя оценить уровень усвоения материала. Связи между таблицами организованы через систему внешних ключей с каскадным удалением, что обеспечивает целостность данных при изменении расписания или удалении пользователей.

Таблица person содержит полные анкетные данные участников учебного процесса, включая ФИО, номер группы и идентификационные данные, что позволяет однозначно соотносить результаты тестирования с конкретными студентами. Особенностью системы является возможность привязки пользователей к учетным записям через связь с таблицей пользователей, что обеспечивает интеграцию процессов тестирования в общую систему управления образовательным процессом. Таким образом, процесс тестирования представляет собой замкнутый цикл от планирования учебных занятий через назначение заданий к фиксации и анализу результатов выполнения.

* + 1. Процесс «Оценка качества проведённого занятия»

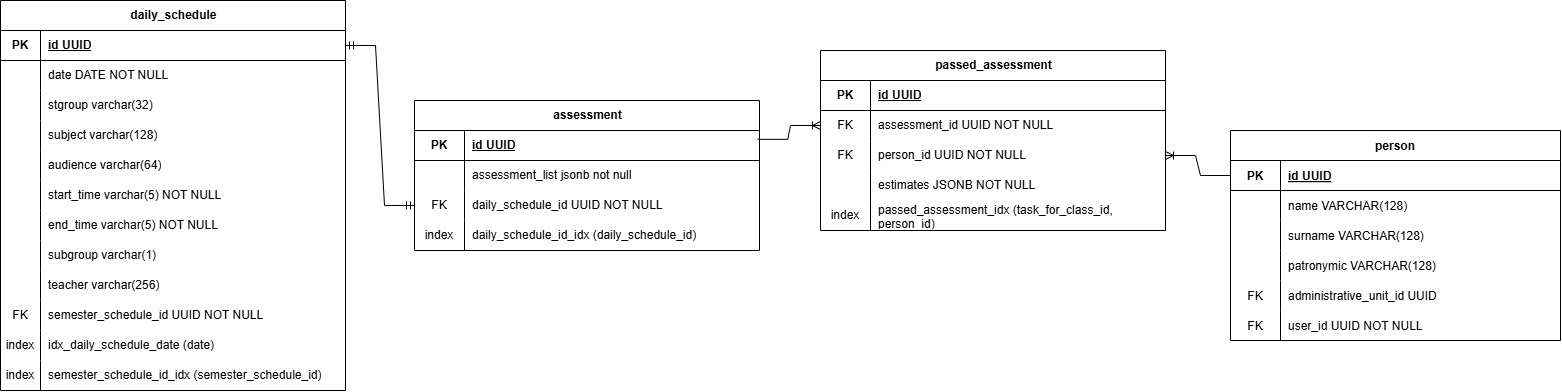


Рис. 2.4 Схема данных процесса «Оценка качества проведённого занятия»

Основой процесса, как и в п. 2.2.3 служит таблица daily\_schedule, содержащая полные данные о запланированных занятиях, включая временные параметры, состав участников и организационные детали. Ключевым элементом оценки является таблица assessment, хранящая в формате JSON набор критериев для предствоящей студентам оценки качества занятия.

Процедура оценки предполагает двустороннюю связь между расписанием и оценочными критериями: с одной стороны, через обязательное поле daily\_schedule\_id в таблице оценок, с другой - через опциональную ссылку assessment\_id в самом расписании. Это позволяет как прикреплять различные оценочные методики к занятиям, так и оперативно находить все оценки для конкретного урока. Результаты оценивания фиксируются в таблице passed\_assessment, где каждый отзыв привязывается к конкретному студенту из таблицы person и содержит детализированные оценки в структурированном JSON-формате.

Особенностью системы является гибкость оценочного механизма: использование JSON-формата позволяет адаптировать критерии оценки под разные дисциплины и типы занятий без изменения структуры базы данных. Каскадное удаление связанных записей обеспечивает целостность данных при обновлении расписания или удалении пользователей. Собранные данные позволяют проводить многомерный анализ качества преподавания, учитывающий как объективные показатели эффективности занятий, так и субъективные оценки участников образовательного процесса.

ДАННЫЕ СТУДЕНТА. МОДУЛЬНЫЙ ЖУРНАЛ

Для того, чтобы обеспечить прозрачность процесса оценки качества занятий, возможность оценивания предоставляется только авторизованным через модульный журнал пользователям. Для этого в системе предусмотрен функционал авторизации через Модульный журнал как OAUTH.2.0-провадйер.

Это означает, что система никак не хранит и не обрабатывает данные о логине и пароле студента. СОКПЗ перенаправит на целевую страницу аутентификации пользователя в модульном журнале, и после успешного прохождения процедуры вернёт токен, с помощью которого можно будет получить актуальные данные студента.

Для этого автор связался с разработчиками данного программного продукта и предоставил данные для возможности такого способа авторизации.

Детально данные процесс будет описан в главе, посвящённой средствам реализации.

ГЛАВА 3. РЕАЛИЗАЦИЯ

ОПИСАНИЕ ТРЕБОВАНИЙ К СРЕДСТВАМ РЕАЛИЗАЦИИ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

Разрабатываемая система, с точки зрения функциональности, должна удовлетворять ряду требований, чтобы выполнять поставленную задачу. Среди таких требований:

* Предоставление удобного пользовательского интерфейса
* Долговременное хранение данных
  + О пользователях и их ролях
  + О занятиях и событиях, связанных с ними
* API (Application Programming Interface) для анализа и агрегации данных и т. д.
* Система аутентификации и авторизации пользователей с ролевой моделью
* Производительная серверная часть

Для каждой, из описанных задач далее будет приведён список средств реализаций, который удовлетворяет требованиям системы.

* + 1. Пользовательский интерфейс

Для инструментов реализации пользовательского интерфейса, в связи с требуемой функциональностью, организовался ряд требований, среди которых:

* Поддержка роутинга «из коробки» - технология должна предоставлять API, для удобного создания многостраничных веб-приложений
* Поддержка асинхронного взаимодействия с сервером – для улучшения пользовательского опыта
* Низкий порог входа

В связи с описанными требованиями, автором было принято решение, реализовывать пользовательский веб-интерфейса используя следующий стек: TypeScript/Next.Js.

TypeScript – расширение языка JavaScript, вводящее ограничения в систему типов языка JavaScript. Благодаря данному расширению контроль над типами производится на этапе компиляции, а ошибки связанные с несовместимостью операций над различными типами, можно выявить на более раннем этапе.

Next.Js – это основанный на React фреймоврк, предназначенный для разработки веб-приложений, обладающих функционалом, выходящим за рамки SPA (Single Page Application), т.е. так называемых одностраничных приложений. Выбор пал в пользу данного инструмента, благодаря встроенной поддержке создания URL на основе исходных путей в файловой системе проекта, что позволяет гибко и просто организовывать иерархию адресов в веб-приложении.

* + 1. Хранение данных

В рамках текущей системы, автором определены три категории данных, которые необходимо хранить и обрабатывать.

* Пользователь и смежные с ним данные (Роли, принадлежность к кафедре, институту, группе и т. п.). Данные определённого формата, которые в дальнейшем с наименьшей вероятностью будут меняться.
* Данные для веб-интерфейса – это созданные самими пользователями структуры данных, для дальнейшего отображения в веб-интерфейсе, например: json-объект содержащий данные для процесса «Тестирвоание».
* Данные о качестве проведённых занятий. Такими данными могут выступать оценки группе/преподавателю или текстовый отзыв.

Для каждой из этих категорий, подобран инструмент, удовлетворяющий требованиям.

* Для хранения данных о пользователе средством реализации будет служить PostgreSQL, т. к. он удовлетворяет ACID (Атомарность, Согласованность, Изолированность, Надёжность). Это означает, что данные в рамках использования PostgreSQL буду согласованы и надёжно сохранены.
* Для хранения данных для рендеринга пользовательского интерфейса будет использован так же PostgreSQL, т.к. его его типы данных json и jsonb обеспечивают необходимую скорость и селективность данных..
* Для того чтобы хранить, агрегировать и анализировать данные, не изобретая свой собственный инструмент, автором принято решение, для этих целей, использовать Elasticsearch. Elasticsearch – поисковой инструмент, позволяющий производительно вести полнотекстовой поиск по данным больших объёмов. Этот механизм реализован благодаря встроенным механизмам токенизациии и обратной индексации. Данный инструмент так же позволяет легко составлять запросы на агрегацию данных различных форматов, что в рамках задачи системы, позволит быстро проводить аналитику.
  + 1. Серверное приложение

К инструменту реализации серверной части, автором описаны следующие требования:

* Статическая типизация со строгой системой типов
* Высокая производительность
* Широкий и доступный набор инструментов, библиотек и фреймворков
* Открытый исходный код

Исходя из описанных требований, в качестве инструментов реализации серверной части системы, автором принято решения использовать следующей стек технологий: Kotlin/Spring Boot/Spring Web/Spring Security/Spring Data JPA/MongoDB/Elasticsearch.

Kotlin – кроссплатформенный, статически типизированный, мульти-парадигменный, работающий поверх JVM (Java Vrtual Machine) язык программирования.

Kotlin позволяет реализовывать серверные приложения, а благодаря поддержке фреймоврка Spring, проекта Spring Boot и его экосистемы, разработка простого веб-приложения может выродиться в написание пары строк кода.

Spring – фреймворк, поддерживающий целый ряд языков семейства JVM, предоставляет инфраструктуру, для написания, как правило, серверных веб-приложений. Spring представляет из себя целую экосистему подпроектов и расширений, которые дополняют уже существующую инфраструктуры совершенно новым функционалом:

* Spring Security – позволяет настроить защиту веб-приложения от несанкционированного доступа.
* Spring Web – расширение, предоставляющее инфраструктуру для создания сетевого взаимодействия между серверами.
* Spring Data \* – расширение, предоставляющее инфраструктуру для взаимодействия с различными СУБД.
* Spring Boot – проект, которому необходимо уделить отдельное внимание. Spring Boot не расширяет функционал, но представляет автоматическую конфигурацию инфраструктуры приложения, встроенный контейнер сервлетов для запуска приложения, и систему стартеров и транзитивных зависимостей, которая позволяет не заботиться о совместимости версий используемых библиотек. Таким образом, загрузив проект из <https://start.spring.io/> , можно фактически сразу запустить веб-приложение.

РЕАЛИЗАЦИЯ

После определения с выбором основных технических средств, автор приступил к разработке целевой системы. На рисунке 3.1 представлена схема архитектуры решения СОКПЗ.

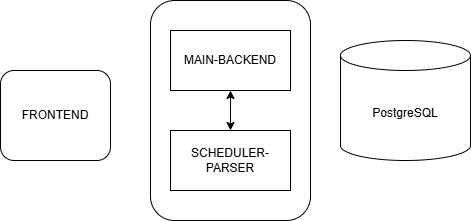


Рис. 3.1 Общая архитектурная схема решения

На данной схеме представлены 4 целевых элемента архитектуры решения:

1. Frontend - пользовательских интерфейс, реализованный с помощью фреймоврка Next.js, языка JavaScript и его диалекта TypeScript.
2. Main-backend - основное серверное приложение. Отвечает за весь функциональный спектр - авторизация, расписание, тестирование и т.д.
3. Scheduler-Parser - дополнительное, но обязательно серверное приложение. Его ответственность - преобразовывать файлы формата pdf в целевую json-структуру для дальнейшего сохранения и отображения расписания.
4. PostgreSQL - целевая СУБД. В ней хранятся фактичски все данные всех процессов системы.

Далее в настоящей главе будут подробно поисаны нюансы и детали реализации каждого компонента

* + 1. Реализация пользовательского интерфейса

Для того чтобы взаимодействие с СОКПЗ стало удобным и интуитивным, автором был разработан пользовательский интерфейс.

Пользовательский интерфейс состоит из ряда различных web-страниц, которые отвечают за определённый функционал для конкретных ролей.

На рисунке 3.2 представлено древо страниц пользовательского интерфейса.

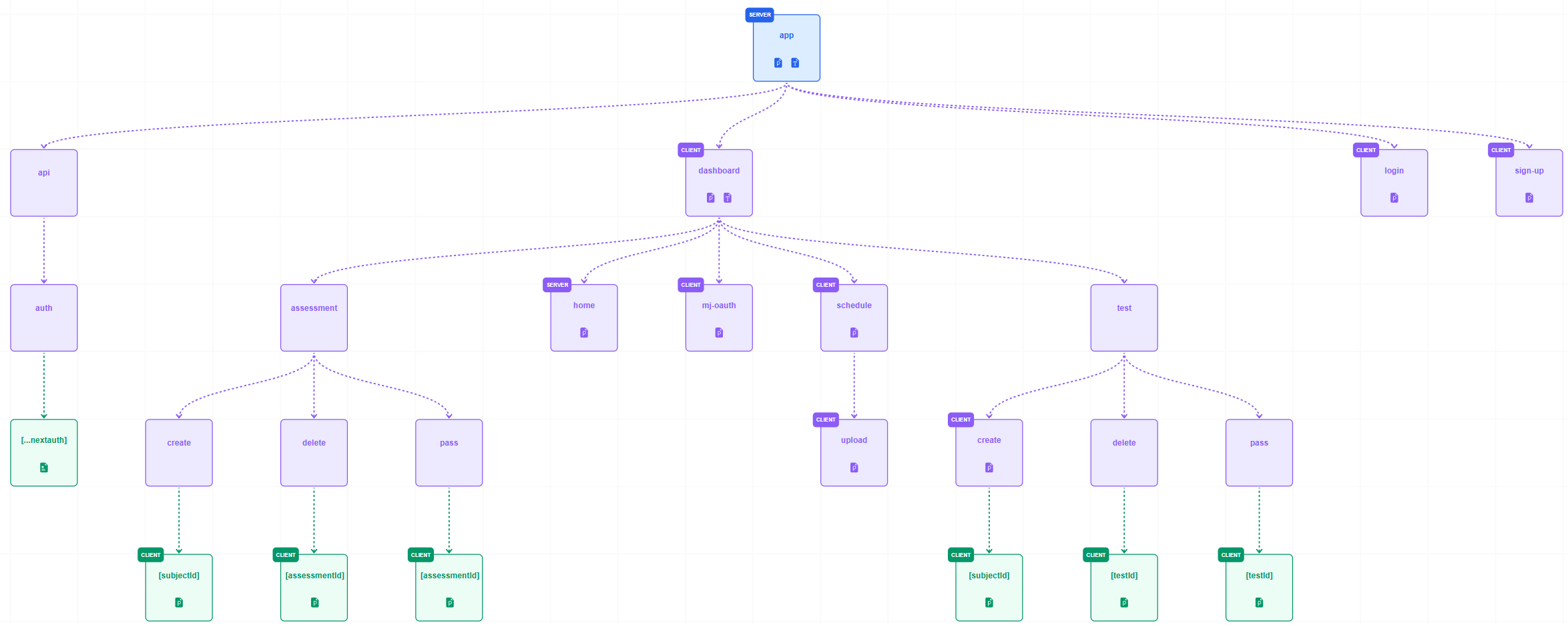


Рис. 3.2 Древо страниц пользовательского интерфейса СОКПЗ

Так же стоит отметить, что в фреймоврке для языка JavaScript Next.js, директория “/app” является основной директорией для организации структуры маршрутов web-страниц. Каждая вложенная директория внутри директории “/app” представляет собой URL-путь. Файлы page.tsx внутри таких директорий - являются непосредственным компилируемым визуальным содержанием web-страницы по относительному директории «/app» адресу пути. Пример такой структуры приведён на рисунке 3.3

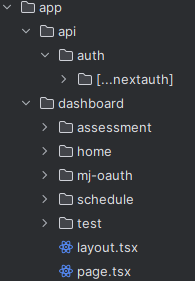


Рис. 3.3 Организационная структура проекта пользовательского интерфейса СОКПЗ

Разобравшись с организационной структурой маршрутов web-страниц, далее в этой главе будут приведены особенности реализации целевой функциональности СОКПЗ.

* + - 1. Расписание занятий

Для того чтобы в пользовательском интерфейсе была возможность удобно создавать и проходить тестирование, проставлять оценки, и совершать удобную навигацию по спискам занятий, автором была разработана система отображения расписания (Рисунок 3.4).

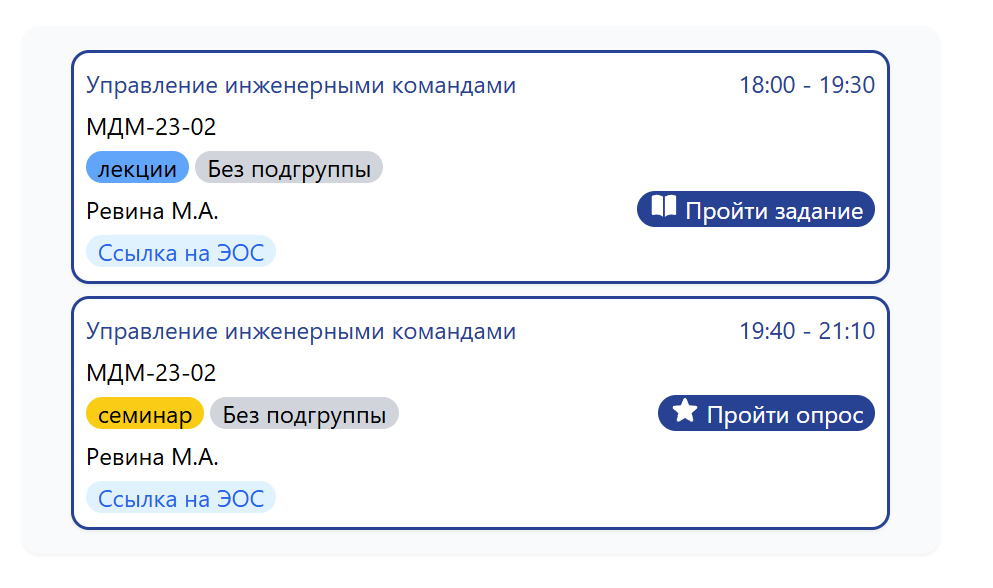


Рис. 3.4 Расписание в пользовательском интерфейсе СОКПЗ

Данный компонент представляет из себя универсальную абстракцию, благодаря которой ученики могут проходить задания преподавателей и их опросы, а преподаватели и внешний аудит управлять (создавать, редактировать, удалять) такие задания и опросы.

В текущем виде предусмотрен один вид создания расписаний - загрузка специального .pdf файла, сгенерированного программой «Ректор-ВУЗ» и поставляемого всем студентам вначале учебного периода. СОКПЗ самостоятельно исключит дубликаты, и уведомит пользователя, что такое расписание уже существует. Форма для загрузки расписания представлена на рисунке 3.5.

В результате успешного заполнения формы, на вспомогательное серверное приложение «scheduler-parser» отправляется целевой pdf-файл с расписанием в бинарном формате для дальнейшего преобразования в целевой формат хранения.

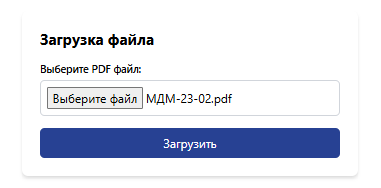


Рис. 3.5 Форма для загрузки расписания в пользовательском интерфейса СОКПЗ

Также в будущем планируется разработка API для синхронизации расписаний с общим репозиторием по определённому cron-таймеру.

Для того, чтобы данные о занятиях в .pdf файле могли быть сохранены в общеупотребимом формате, система СОКПЗ сначала создает структуру в формате JSON, разворачивает данный JSON в список занятий, сортирует занятия и сохраняет в реляционную базу данных. Подробней данный процесс будет рассмотрен в главе, посвящённой средствам реализации.

* + - 1. Тестирование и опросы

На рисунке 3.5 представлен компонент расписания, доступный для роли «Преподаватель» и «Аудитор» для создания элементов заданий и опросов.

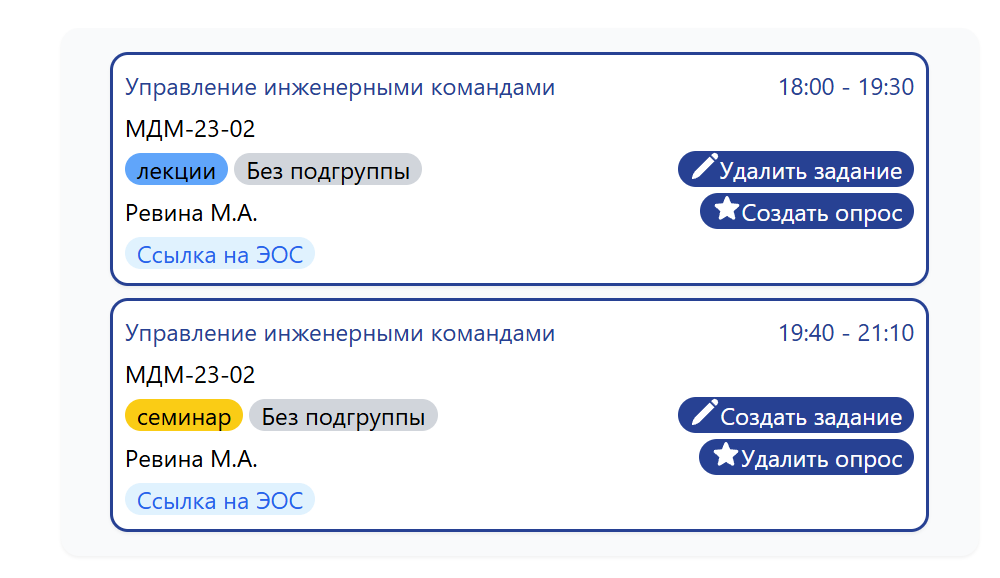


Рис. 3.6 Управление тестами и опросами

На рисунке 3.7 представлена форма для создания элементов тестирования. Данная форма позволяет задать необходимое количество вопросов с различным количеством вариантов ответов. В результате успешного заполнения формы на сервер отправляются данные о тестировании, которые в дальнейшем отобразятся у группы, которой принадлежит данный элемент ежедневного расписания в соответствии с сформированной схемой данных.

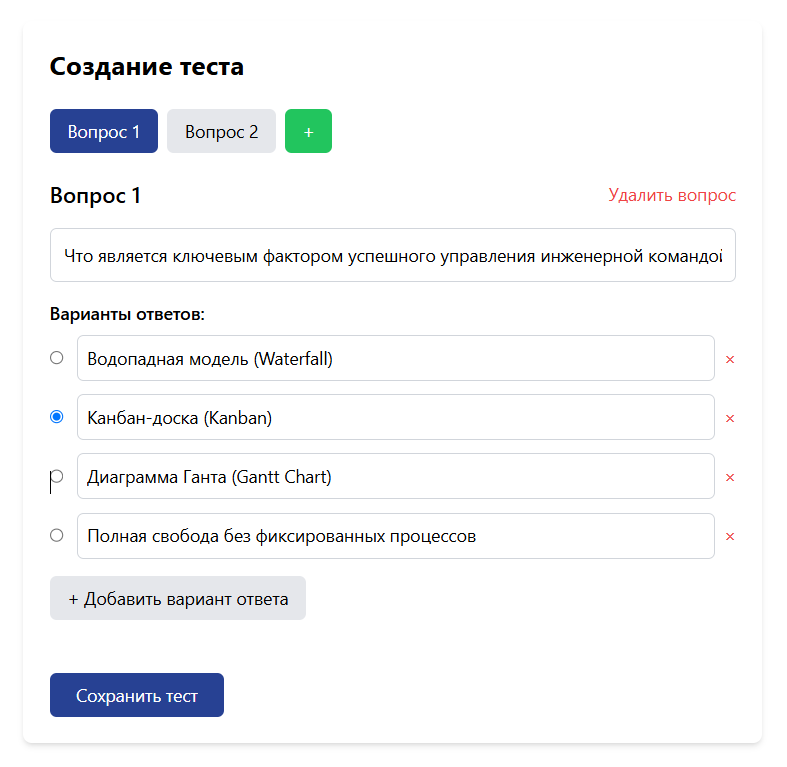


Рис. 3.7 Форма создания элемента тестирования

На рисунке 3.8 представлена форма для создания элементов опроса об оценке качества проведённого занятия. Данная форма позволяет задать необходимое количество вопросов, ответом на которые должна быть целочисленная оценка по 10-бальной шкале. В результате успешного заполнения формы на сервер отправляются данные об опросе, которые в дальнейшем отобразятся у группы, которой принадлежит данный элемент ежедневного расписания в соответствии с сформированной схемой данных.

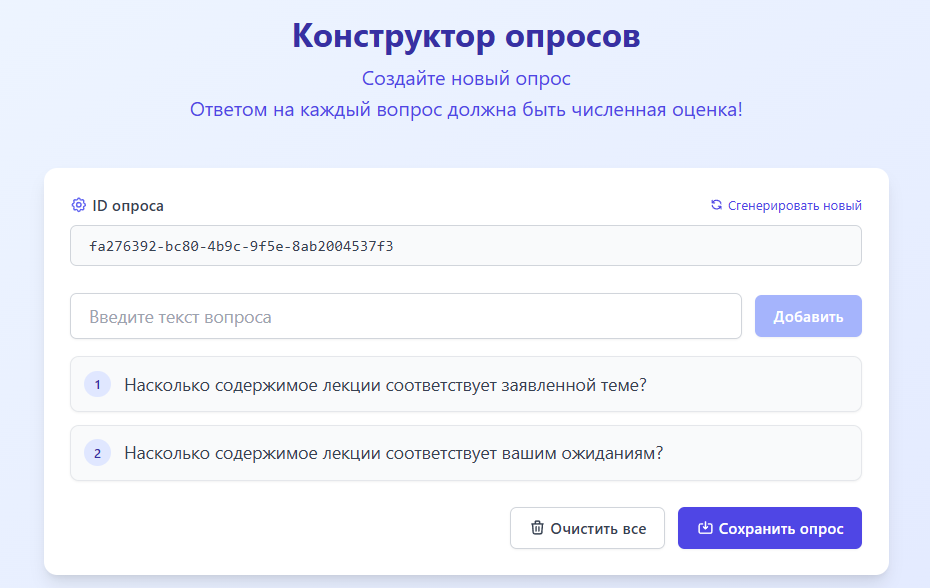


Рис. 3.8 Форма создания элемента опроса оценки качества проведённого занятия

На рисунке 3.9 представлена форма для прохождения задания, созданного ролью «Преподаватель». Данная форма позволяет численно оценить качество и уровень вовлеченности у роли «Студент» и сохранить полученные данные для дальнейшей агрегации. В результате успешного заполнения формы на сервер отправляются данные об ответах студента, в частности ссылка на опрос, ссылка на студента, выполнявшего задание и численный результат.

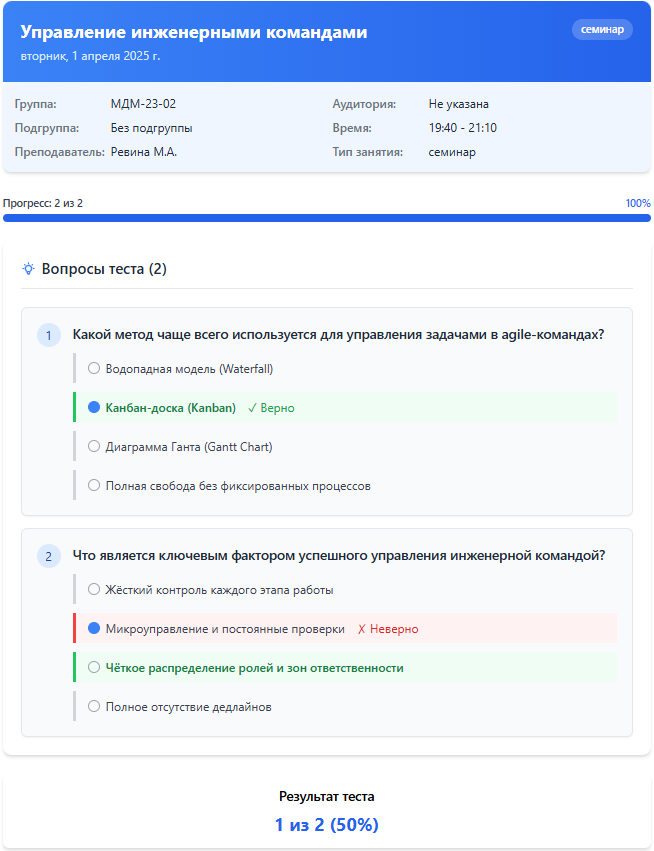


Рис. 3.9 Форма прохождения элемента тестирования ролью «Студент»

На рисунке 3.10 представлена форма для сбора оценок качества проведённого занятия. Данная форма позволяет численно оценить субъективную оценку роли «Студент» качества проведённого занятия. В результате успешного заполнения формы на сервер отправляются данные оценок в 10-бальной шкале, ссылка на опрос и ссылка на студента, выполнявшего задание.

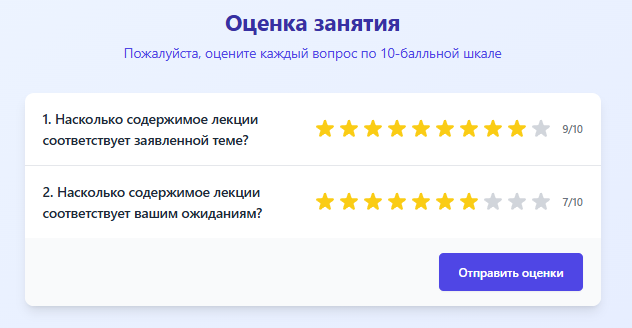


Рис. 3.10 Форма прохождения элемента опроса об оценке качества проведённого занятия

Для удаления элементов тестирования и опросов так же предусмотрены

формы, в результате успешного заполнения которых отправляется запрос на сервер с целью удаления из базы данных таких сведений. Примеры форм приведены на рисунках 3.11 и 3.12.

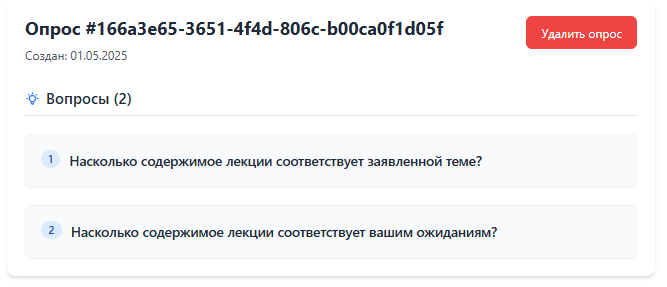


Рис. 3.11 Форма удаления элемента опроса из базы данных

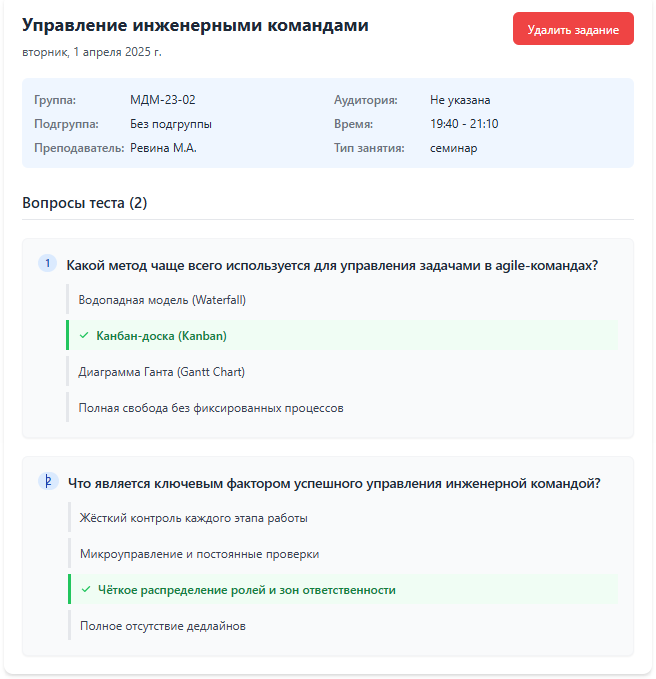


Рис. 3.12 Форма удаления элемента тестирования из базы данных

* + - 1. Модульный журнал. OAUTH 2.0 провайдер

Для того чтобы однозначно идентифицировать сущность студента и привязать данные студента к сущности пользователя системы, автором системы была разработана интеграция с модульным журналом «МГТУ СТАНКИН».

Сервис «Модульный журнал» может быть использован для как средство аутентификации студентов МГТУ Станкин и получения их персональных данных, таких как имя, фамилия, номер студенческого билета, группы и т.д. посредством протокола OAuth 2.0.

На данный момент автоматическая регистрация сторонних сервисов в системе «Модульный журнал» не реализована. По этой причине автор связался с главным разработчик для регистрации СОКПЗ.

Процедура аутентификации выглядит следующим образом:

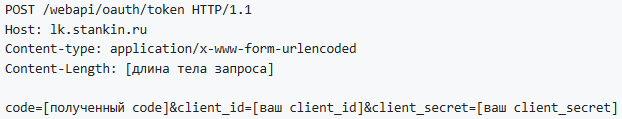
1. На сайте необходимо разместить ссылку «https://lk.stankin.ru/webapi/oauth/authorize?response\_type=code&client\_id=[client\_id]&redirect\_uri=[redirect\_uri]», где:
   1. [client\_id] - идентификатор интегрируемой системы, в случае СОКПЗ - rate-please-stankin
   2. [redirect\_uri] - адрес сервера пользовательского интерфейса, на который необходимо вернуть управление.
2. При клике на ссылку из п.1 пользователь будет перенаправлен на страницу запроса разрешения передачи данных интегрируемому сервису.
3. В случае введения корректных логина и пароля студента, пользователь будет перенаправлен на [redirect\_url], при этом к строке запроса будет добавлен параметр [code], при помощи которого появится возможность получить информацию о пользователе.
4. Пользовательский интерфейс отправляет запроса на «main» серверное приложение, которое в свою очередь отправляет запрос в систему «Модульный журнал». Пример запроса приведён на рисунке 3.13
5. В ответ пользовательский интерфейс получает данные студента. Пример данных приведён на рисунке 3.14  
   

Рис 3.13 Форма удаления элемента тестирования из базы данных



Рис. 3.14 Пример данных студента, полученнёх из сервиса «Модульный журнал»

В результате прохождения процедуры аутентификации, в пользовательском интерфейсе отобразятся данные студента (рисунок 3.15). Так же в данной форме доступна кнопку привязывания данных студента с сущностью пользователя системы СОКПЗ.

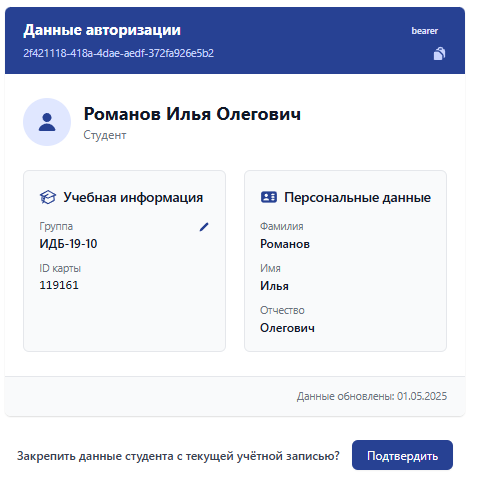


Рис. 3.15 Пример данных студента, полученных из сервиса «Модульный журнал»

Без выполнения данной процедура, пользователю не удастся выполнить тестирование и пройти опрос контроля качества проведённых занятий, что обеспечивает прозрачность процесса.

Также в данном компоненте доступна кнопка изменения данных о группе студента, т.к. существует ситуации, в которых такие сведения могут быть неактуальны, а расписание отображается только для группы, указанной в данной форме.

* + 1. Реализация серверной части

Серверная часть СОКПЗ имеет два серверных приложения:

1. Основное серверное приложение - представляет из себя монолит, имеющий REST API для интеграции с пользовательским интерфейсом. В нём реализованы HTTP - конечные точки, отвечающие за авторизацию и аутентификацию, управление расписанием, тестированием и опросами, а так же генерацией целевых отчетов.
2. Дополнительное серверное приложение - является утилитарным и предназначено для преобразования целевого файла формата pdf в структуру данных формата json с общими сведениями о расписании. После преобразования, сервис отправляет полученные данных в основное серверное приложение, где затем происходит финальное преобразование и сохранение в базу данных.

Разобравшись с общей функциональной принадлежностью, далее будут описаны особенности реализации серверной части СОКПЗ