

РАЗРАБОТКА СИСТЕМЫ ТЕСТИРОВАНИЯ ДЛЯ ОЦЕНКИ И КОНТРОЛЯ ЗНАНИЙ

Харионовский Константин Михайлович¹, Поздеев Сергей Витальевич²,
Матвеев Дмитрий Алексеевич³, Епишина Екатерина Викторовна⁴

¹Разработчик;

ООО «Клауд Ком»;

Россия, 141983, Московская область, г. Дубна, ул. Программистов, 4;

e-mail: kostya1337228322@yandex.ru.

²Разработчик;

ООО «Клауд Ком»;

Россия, 141983, Московская область, г. Дубна, ул. Программистов, 4;

e-mail: pozdeev.2002@inbox.ru.

³Разработчик;

ООО «Клауд Ком»;

Россия, 141983, Московская область, г. Дубна, ул. Программистов, 4;

e-mail: jeazzly@mail.ru.

⁴Старший преподаватель;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: e.epishina@nordavind.ru.

В статье исследованы функционал и применяемые технологии в современных системах тестирования; проанализированы системы тестирования оценки знаний Moodle, Mirapolis LMS, Teachbase, и WebTutor; сформулированы требования к системе тестирования, различные сценарии взаимодействия и дизайн системы, описаны основные модули системы.

Ключевые слова: электронное обучение, LMS-системы, образование, тестирование, мессенджеры.

Для цитирования:

Разработка системы тестирования для оценки и контроля знаний / К. М. Харионовский, С. В. Поздеев, Д. А. Матвеев, Е. В. Епишина // Системный анализ в науке и образовании: сетевое научное издание. 2024. № 4. С. 45-56. EDN: ИНОГВО. URL: <https://sanse.ru/index.php/sanse/article/view/637>.

DEVELOPMENT OF A TESTING SYSTEM FOR ASSESSMENT AND CONTROL OF KNOWLEDGE

Kharionovskiy Konstantin M.¹, Pozdeev Sergey V.²,
Matveev Dmitriy A.³, Epishina Ekaterina V.⁴

¹Developer;

LLC "Cloud Com";

4 Programmistskaya Str., Dubna, Moscow region, 141983, Russia;

e-mail: kostya1337228322@yandex.ru.

²Developer;

LLC "Cloud Com";

4 Programmistskaya Str., Dubna, Moscow region, 141983, Russia;

e-mail: pozdeev.2002@inbox.ru.



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

³Developer;

LLC "Cloud Com";

4 Programmistskaya Str., Dubna, Moscow region, 141983, Russia;

e-mail: jeazzly@mail.ru.

⁴Senior teacher;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: e.epishina@nordavind.ru.

The article examines the functionality and applied technologies in modern testing systems; the knowledge assessment testing systems Moodle, Mirapolis LMS, Teachbase, and WebTutor are analyzed. the requirements for the testing system are formulated, various scenarios of interaction and system design are described, the main modules of the system are described.

Keywords: e-learning, LMS systems, education, testing, messengers.

For citation:

Kharionovskiy K. M., Pozdeev S.V., Matveev D. A., Epishina E. V. Development of a testing system for assessment and control of knowledge, *System analysis in science and education*, 2024;(4):45-56 (in Russ). EDN: ИНОГБО. Available from: <https://sanse.ru/index.php/sanse/article/view/637>.

Введение

В современном мире непрерывно растут требования к качеству образования и необходимость внедрения современных технологий в учебный процесс. Разработка системы тестирования становится актуальной задачей, способствующей повышению эффективности обучения и контроля знаний. Электронные системы тестирования позволяют упростить процесс подготовки, проведения и анализа результатов тестов для преподавателей, а также обеспечивают возможность оперативной обратной связи для студентов.

1. Анализ

Для выявления ключевых трендов и определения требований для разработки системы тестирования было проведено исследование современных систем тестирования в образовательной сфере. Анализ решений проводился на основе комплекса критериев, включающих в себя:

- функциональное наполнение: разнообразие типов вопросов и модулей тестирования, настройки сложности, индивидуализация процесса обучения;
- гибкость настройки: возможность адаптации системы под специфические нужды учебного заведения или курса, в том числе через добавление собственных материалов и инструментов аналитики;
- интеграция: совместимость с другими образовательными платформами и информационными системами, облегчающая обмен данными и расширение функционала;
- пользовательский интерфейс: интуитивность и удобство использования как для администраторов, так и для конечных пользователей;
- масштабируемость и производительность: способность системы эффективно работать с различным объемом пользователей и данных;
- безопасность и конфиденциальность данных: защита личной информации и результатов тестирования от несанкционированного доступа.

В качестве объектов исследования были выбраны ключевые платформы, занимающие лидирующие позиции на рынке систем тестирования в образовательной сфере: *Moodle*, *Mirapolis LMS*, *Teachbase*, и *WebTutor*. Для проведения анализа использовались как открытые данные из доступных Интернет-источников, так и информация, полученная через использование демонстрационных версий. На основе собранных данных проведено сравнение по заранее определенным критериям. Краткие результаты данного исследования представлены в виде табл. 1.

Табл. 1. Результаты сравнительного анализа

Критерии/Решения	<i>Moodle</i>	<i>Mirapolis LMS</i>	<i>Teachbase</i>	<i>WebTutor</i>
Функциональное наполнение	+	+	-	+
Гибкость настройки	+	-	-	-
Интеграция	+	+	+	-
Пользовательский интерфейс	-	+	+	-
Масштабируемость и производительность	-	-	+	+
Безопасность и конфиденциальность данных	+	+	+	+

В рамках сравнительного анализа функциональных возможностей популярных платформ для управления обучением можно выделить несколько ключевых аспектов. *Moodle* – открытая платформа с широким набором плагинов, предлагает высокую гибкость и интеграцию с различными системами, однако может столкнуться с проблемами масштабирования. *Mirapolis LMS*, ориентированная на корпоративный сектор, обеспечивает высокий уровень защиты данных, хотя требует больше ресурсов для оптимизации. *Teachbase* предлагает современный дизайн и поддержку мобильных устройств, но может иметь ограниченные возможности интеграции. *WebTutor*, специализирующийся на корпоративном обучении, подходит для крупных организаций, но требует значительных усилий для настройки и интеграции. Каждая из этих платформ имеет свои уникальные преимущества и ограничения, что важно учитывать при выборе системы для конкретных образовательных или корпоративных задач.

В результате проведенного сравнительного анализа систем *Moodle*, *Mirapolis LMS*, *Teachbase* и *WebTutor*, основанного на функциональных возможностях, преимуществах и недостатках каждого из рассмотренных решений, можно сделать вывод о том, что, несмотря на их значительные сильные стороны и широкий спектр предлагаемых возможностей, ни одна из этих платформ не удовлетворяет всем требованиям и специфическим потребностям нашего проекта в полной мере.

На основе проведенного анализа существующих решений были составлены требования, которым должна удовлетворять система тестирования качества обучения. Ниже представлены основные требования.

Функциональные требования:

1. Создание тестов. Преподаватель должен иметь возможность создавать тесты.
2. Распределение тестов. Преподаватель должен иметь возможность распределять тесты по группам студентов.
3. Управление группами студентов. Преподаватель должен иметь возможность объединять студентов в учебные группы.
4. Мониторинг. Преподаватель должен иметь возможность отслеживать результаты тестирования студентов.

Требования к данным:

В системе должна храниться информация о студентах, группах, тестах, дисциплинах, вопросах, ответах и остальных исходных данных.

Требования к защите данных:

Для обеспечения защиты данных должен быть предусмотрен механизм аутентификации для разных групп пользователей (супер-админ, админ и преподаватель).

2. Архитектура системы

При разработке системы тестирования за основу была взята клиент-серверная архитектура. Данная архитектура позволяет реализовать распределенную обработку, поскольку часть выполняется на стороне клиента, а часть - на стороне сервера. Это позволяет снизить нагрузку сервера и оптимизировать его работу, а также увеличить число пользователей, одновременно работающих с сервером. Клиентская часть состоит из двух компонентов. Телеграм-бот – в качестве пользовательского интерфейса для студентов, для преподавателей – Web приложение, разработанное на *JavaScript* с использованием библиотеки *React*. Клиентская часть взаимодействует с серверным приложением через *API* по *HTTP* протоколу. Серверное приложение отвечает за логику взаимодействия с базой данных, обработку запросов и управление всеми аспектами системы. Серверная часть написана на языке *Java* с использованием фреймворка *Spring Boot*. Для хранения всей информации используется PostgreSQL. На рис. 1 схематично изображена архитектура системы.

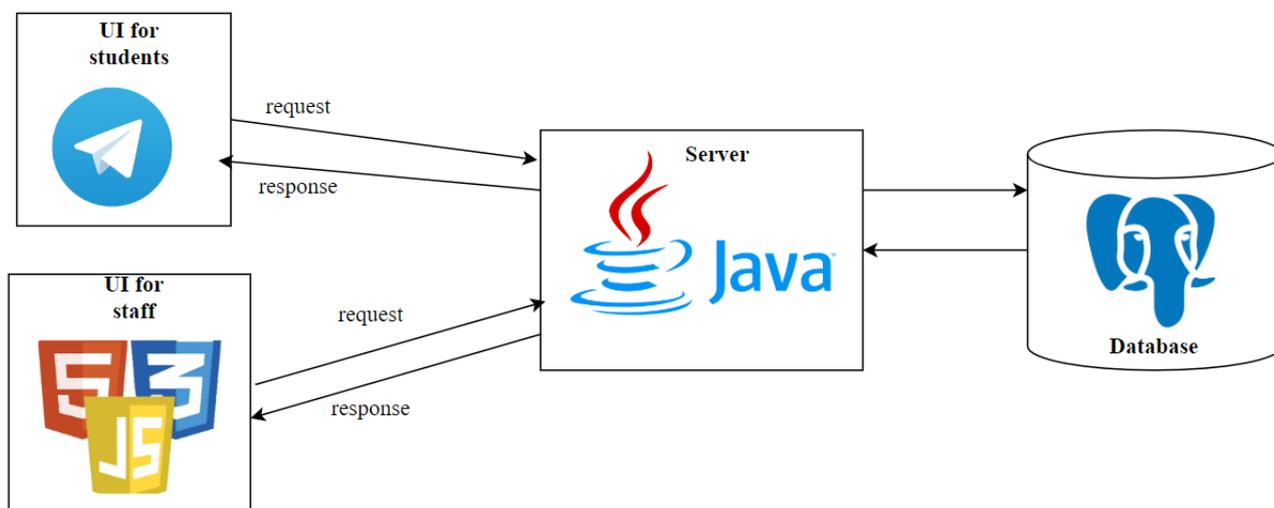


Рис. 1. Архитектура системы

3. Функциональная модель системы

Для того чтобы более наглядно представить функциональные возможности и взаимодействия в системе тестирования, были разработаны диаграммы вариантов использования (*Use-case* диаграммы). Эти диаграммы позволяют визуализировать основные функции системы и их взаимосвязи с различными категориями пользователей.

Супер-администратор имеет расширенные права и возможности по сравнению с обычными пользователями и администраторами, сценарии использования для роли супер-администратора продемонстрированы на рис. 2.

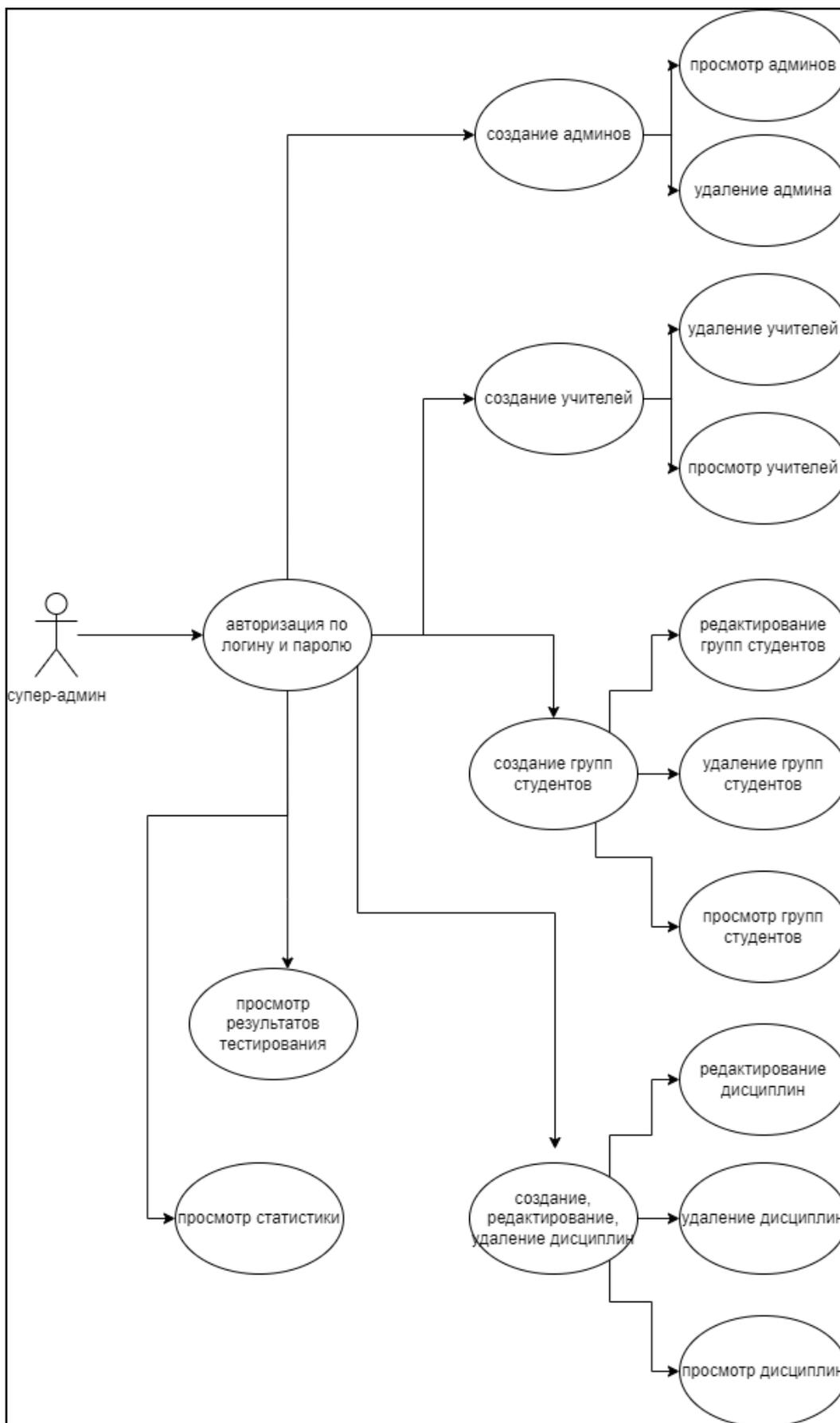


Рис. 2. Use-case диаграмма для роли супер-администратор

Администратор обладает полномочиями, позволяющими ему эффективно управлять процессом тестирования и поддерживать работоспособность системы. Основные возможности администратора проиллюстрированы в формате *Use-case* диаграммы на рис. 3.

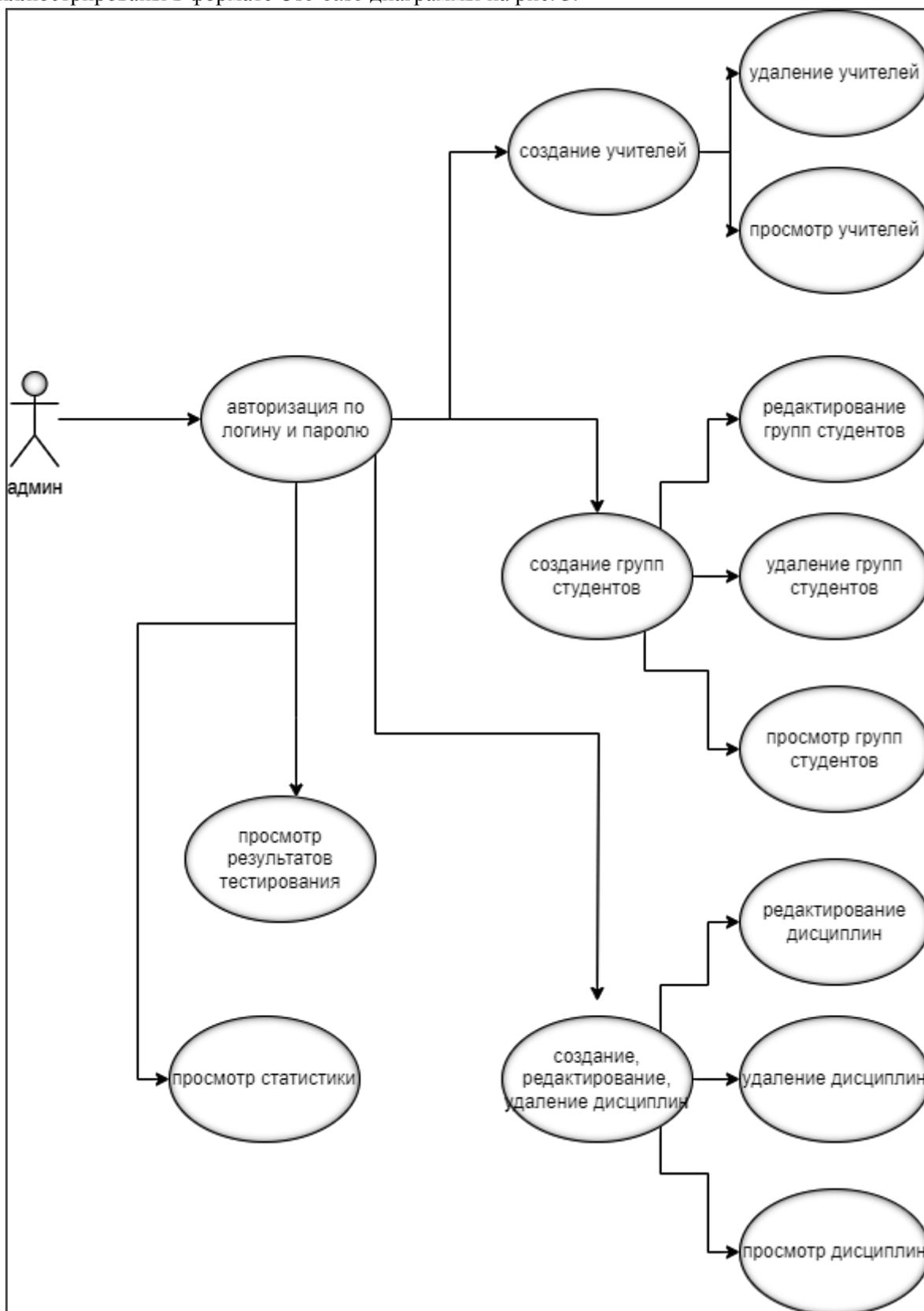


Рис. 3. Use-case диаграмма для роли администратор

На рис. 4 изображены возможные сценарии взаимодействия с системой для преподавателя. Диаграмма отражает возможности создания и управления тестами, мониторинга результатов студентов, а также предоставления обратной связи.

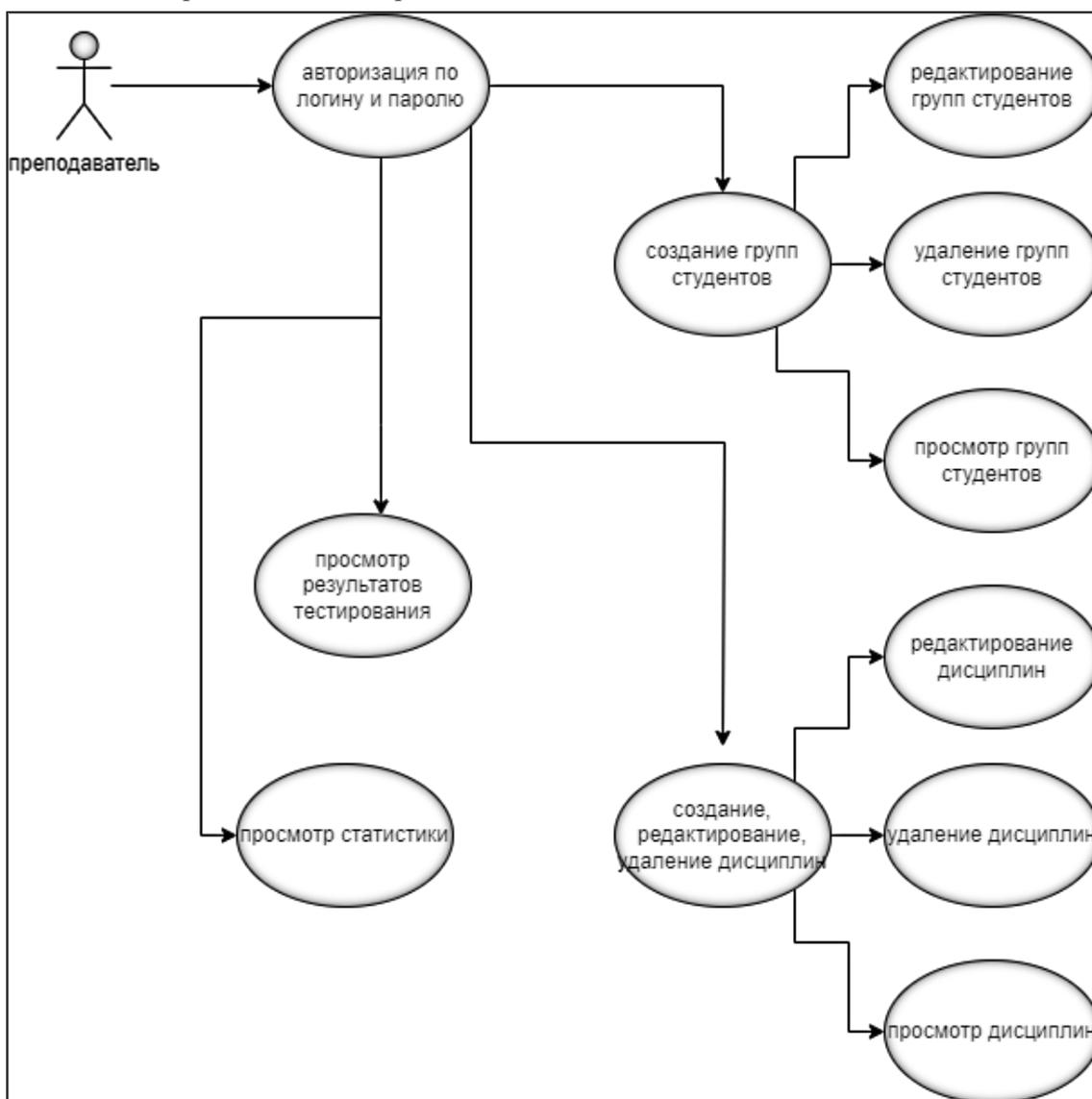


Рис. 4. Use-case диаграмма для роли преподаватель

Студент взаимодействует с системой через мессенджер. При помощи *Telegram*-бота он может зарегистрироваться, пройти тест и посмотреть результат теста, эти сценарии проиллюстрированы на рис.5.

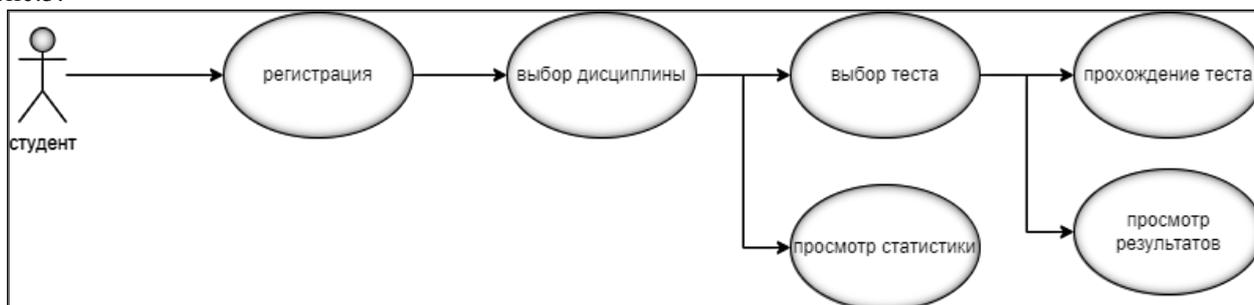


Рис. 5. Use-case диаграмма для роли студент

4. Разработка интерфейса

В работе были спроектированы и реализованы все сущности, необходимые для полноценной работы системы. При первом входе пользователь попадает на страницу авторизации (см. рис. 6).

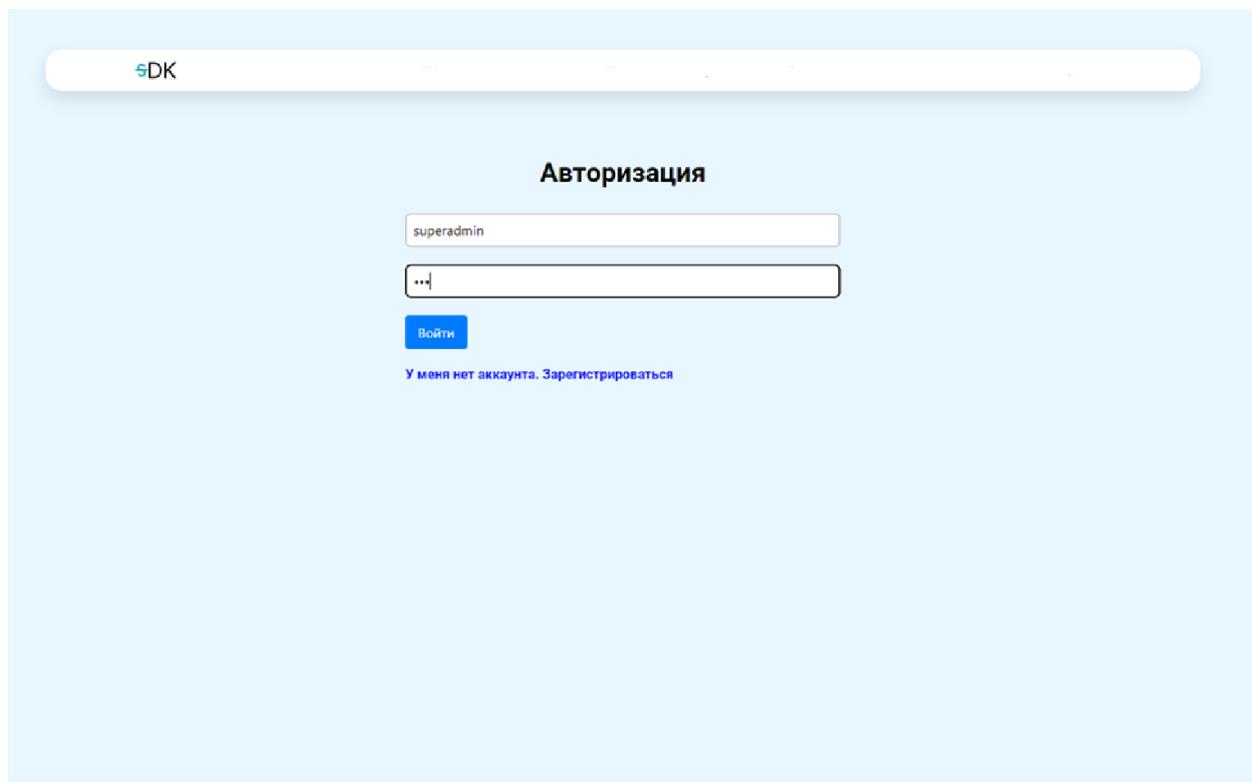


Рис. 6. Страница авторизации

После удачной авторизации пользователь видит доступный ему функционал. В верхней части экрана расположена навигационная панель, на которой находятся вкладки для перехода между основными разделами системы. Это обеспечивает интуитивно понятное управление и навигацию. На странице дисциплин можно создать новую дисциплину, удалить существующую и редактировать дисциплину (см. рис. 7).

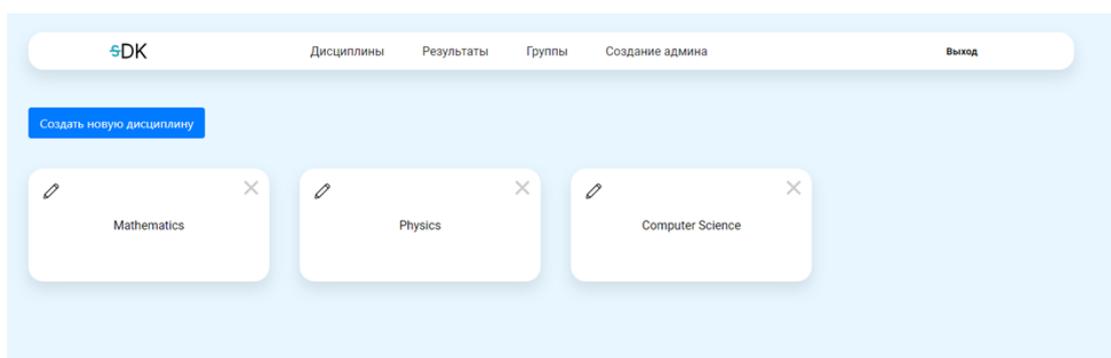


Рис. 7. Страница дисциплин

На вкладке «Группы» преподаватель имеет возможность создавать группы студентов, редактировать их и прикреплять к ней дисциплины (см. рис. 8).

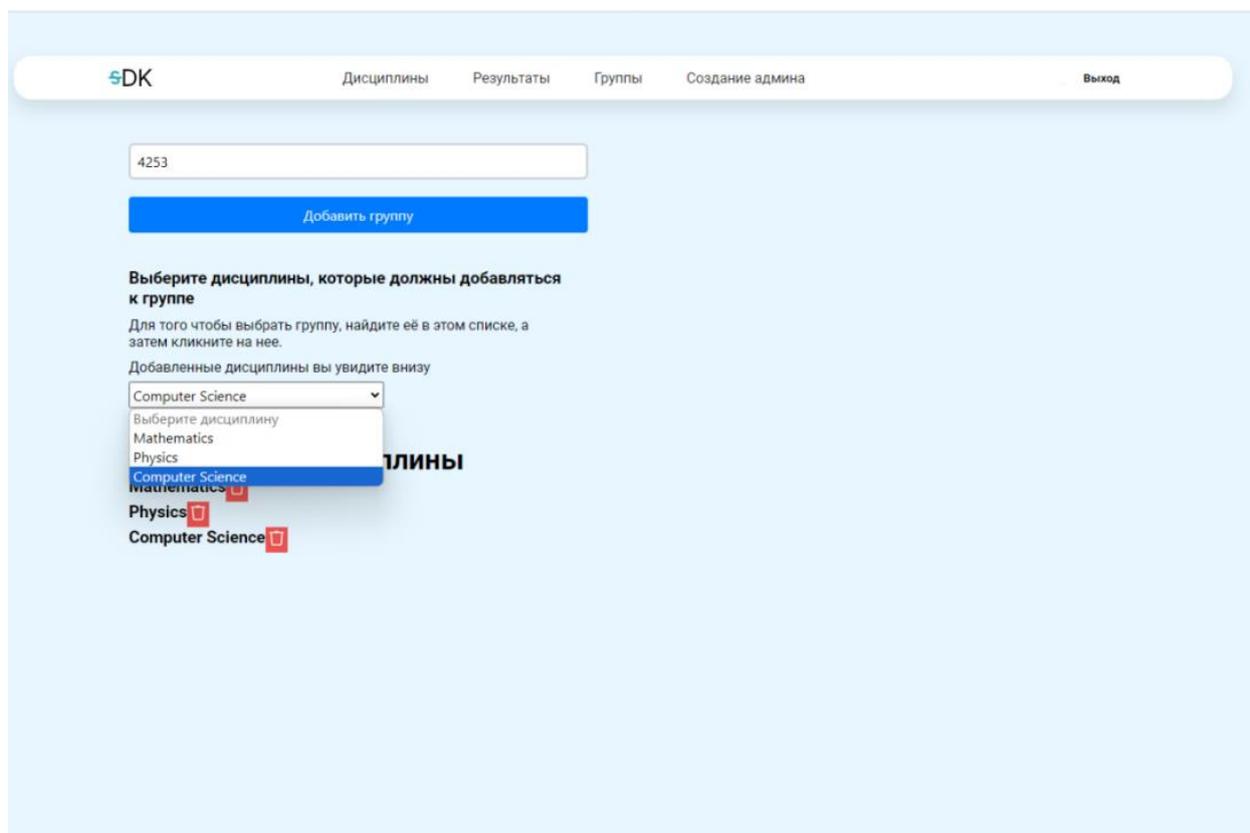


Рис. 8. Страница дисциплин

На четвертой вкладке веб-страницы, предназначенной для администрирования, осуществляется управление различными ролями пользователей, такими как *ROLE_SUPERADMIN*, *ROLE_ADMIN*, *ROLE_TEACHER* и *ROLE_USER*, согласно представленной ранее иерархии. Для добавления администратора необходимо указать электронную почту, пароль и осуществить нажатие на кнопку "Создать". Также на данной странице можно управлять уже существующими пользователями: удалять их или изменять их права доступа (см. рис. 9).

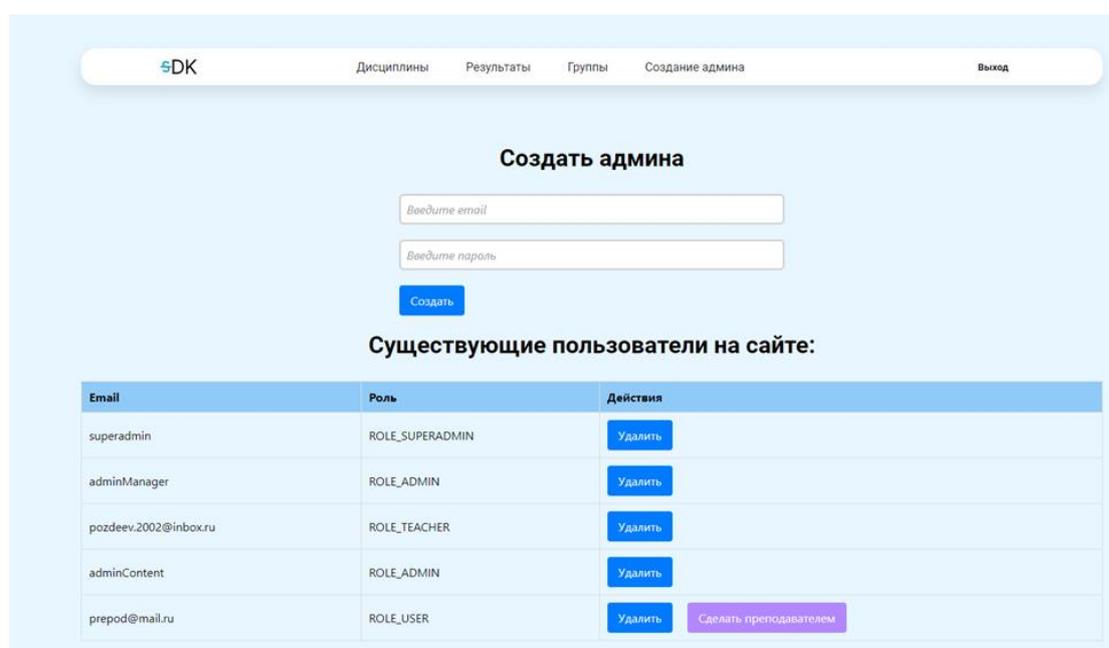


Рис. 9. Управление ролями

Преподаватель имеет возможность просмотреть результаты теста, ФИО ученика, который проходил тест в *Telegram*-боте, название теста, и процент правильных ответов (см. рис. 10).

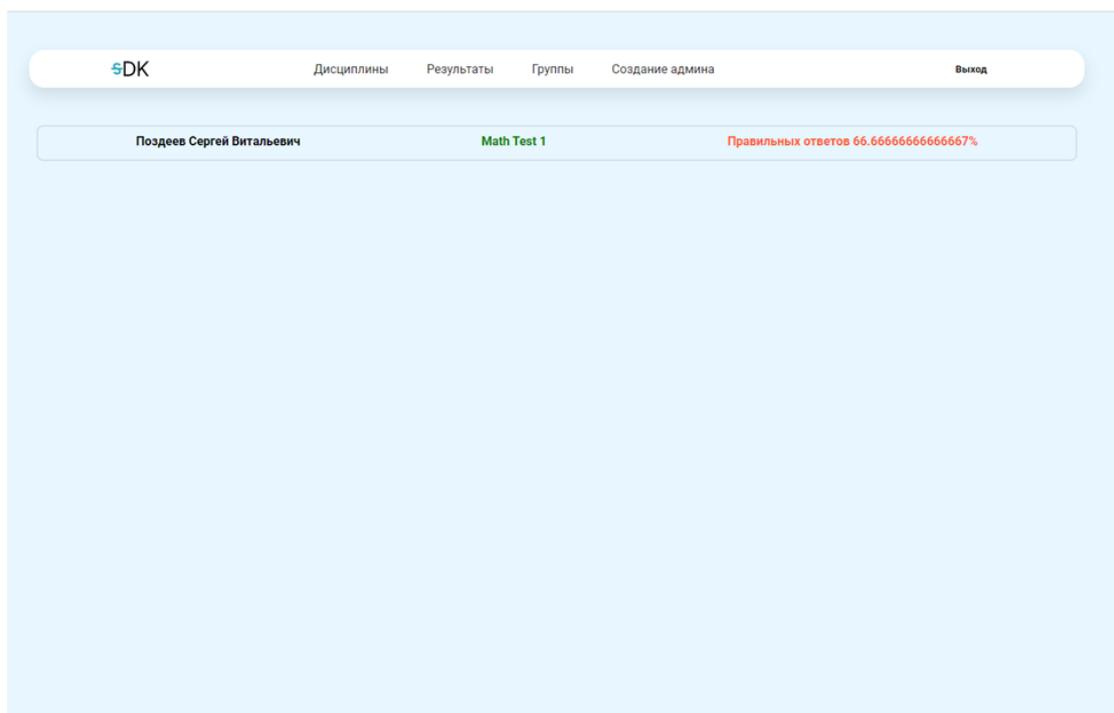


Рис. 10. Страница просмотра результатов

При первом взаимодействии с ботом в *Telegram* пользователь должен написать `/start` и следовать инструкции (см. рис. 11).

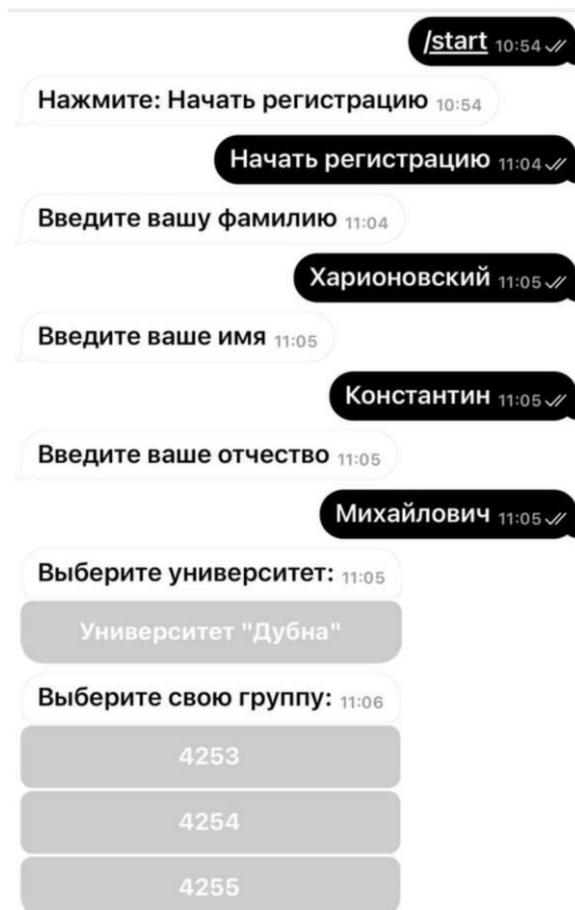


Рис. 11. Регистрация

После успешной регистрации ему будет доступно меню, действия в котором осуществляются посредством нажатия на кнопки. Далее показан стандартный сценарий взаимодействия студента с системой (см. рис. 12-14).



Рис. 12. Выбор теста



Рис. 13. Прохождение теста

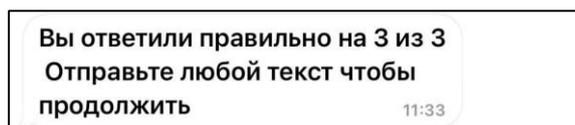


Рис. 14. Результат теста

Заключение

В ходе работы был проведен анализ существующих систем тестирования и выявлены их ключевые функциональные и нефункциональные требования. На основе этих требований было спроектировано и разработано собственное решение. При реализации за основу выбрана клиент-серверная архитектура.

В процессе проектирования системы были созданы диаграммы вариантов использования для визуализации функциональных возможностей и взаимодействий пользователей с системой. Также была разработана диаграмма последовательности для моделирования взаимодействия объектов и последовательности обмена сообщениями между ними.

Реализованы основные компоненты системы: серверная часть, клиентская часть, *Telegram*-бот и база данных. Для хранения и управления кодом использовалась система версионирования *Git*, а репозиторий проекта был размещен на платформе *Github*. После инсталляции системы был проведен процесс тестирования, который подтвердил её работоспособность и соответствие требованиям.

Разработанное решение обеспечивает эффективное и удобное тестирование, взаимодействие преподавателей и студентов, а также предоставляет средства для мониторинга и анализа результатов.

Список источников

1. Иванов А. Н., Власюк П. С. Разработка приложения с использованием веб-технологий Java // Ученые заметки ТОГУ. – 2020. – Т. 11, № 4. – С. 261-266.
2. Blosh J. Effective java – Addison-Wesley Professional, 2017. – 416 с.
3. Яценко А. Фреймворк Spring: зачем он нужен, как устроен и как работает // Skillbox.ru — журнал для профессионалов. Актуальные статьи. – Skillbox, 2024 – Дата публикации: 06.07.2022. – URL: <https://skillbox.ru/media/code/freymvork-spring-zachem-on-nuzhen-kak-ustroen-i-kak-rabotaet/> (дата обращения: 21.04.2024).
4. PostgreSQL : Документация... : Компания Postgres Professional // Postgres Professional — Компания Postgres Professional. – URL : <https://postgrespro.ru/docs/postgresql/9.6/app-postgres> (дата обращения: 21.04.2024).
5. Git : [распределенная система управления версиями с открытым исходным кодом]. – URL: <https://git-scm.com/book/ru/v2> (дата обращения: 21.04.2024).
6. Setting Up Swagger 2 with a Spring REST API // Baeldung. – Дата обновления: 05.07.2023. –URL: <https://www.baeldung.com/swagger-2-documentation-for-spring-rest-api> (дата обращения: 25.04.2024).
7. Getting Started — React // React — A JavaScript library for building user interfaces. – Meta Platforms, Inc.¹, 2024. – URL: <https://legacy.reactjs.org/docs/getting-started.html> (дата обращения: 11.05.2024).
8. Feature Overview // React Router Official Documentation. – Shopify, Inc., 2024.– URL: <https://reactrouter.com/en/main/start/overview/> (дата обращения: 20.05.2024).

¹ Деятельность Meta Platform Inc. на территории РФ запрещена.