

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
**«Белгородский государственный технологический университет им. В.Г.
Шухова»**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**«Разработка front-end Web-приложения – учебной среды с чатами и
AI-анализом кода лабораторных работ»**

Автор работы: Бондаренко Сергей Владимирович

Направление подготовки: 09.03.04 «Программная инженерия»

Руководитель: Мельников Антон Борисович, руководитель департамента автоматизации бизнеса ООО «Технологии надежности»

Цель и задачи

Цель: повысить эффективность и облегчить работы преподавателей и обучения студентов через создание front-end части Web-приложения для управления учебным процессом, общения и автоматической проверки заданий.

Задачи:

- 1 Провести анализ существующих образовательных решений.
- 2 Определить архитектуру Web-приложения и технологический стек.
- 3 Разработать пользовательский интерфейс.
- 4 Реализовать модули для управления учебными структурами, заданиями и системой общения.
- 5 Интегрировать модуль автоматической проверки решений с использованием ИИ.
- 6 Провести тестирование бизнес-логики приложения.

Анализ существующих образовательных решений

Проблема заключается в том, что Google Classroom, Microsoft Teams for Education и Moodle решают задачи управления заданиями. VK, Telegram и Viber предназначены только для общения. А CodeSignal и Codility реализуют исключительно AI-анализ кода.

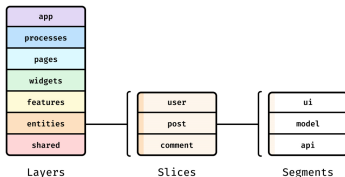
Из-за разделения функций между разными сервисами преподаватели и студенты вынуждены постоянно переключаться между несколькими приложениями, что создаёт неудобства и снижает эффективность работы.

Аналоги/Функции	Система заданий	Общение	AI-анализ
Google Classroom	+	—	—
MS Teams	+	+	—
Moodle	+	—	—
CodeSignal	—	—	+
Codility	—	—	+

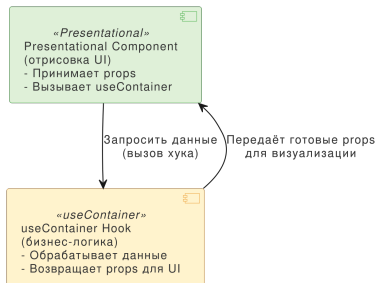


- **Next.js** — используется для серверного рендеринга (SSR), маршрутизации и повышения производительности клиентского приложения.
- **React** — обеспечивает декларативный подход к построению компонентного пользовательского интерфейса.
- **TypeScript** — добавляет статическую типизацию, улучшает читаемость и сопровождаемость кода.

Архитектура Web-приложения



FSD-архитектура



Контейнерно-презентационный подход

- **Feature-Sliced Design (FSD)** — архитектурный подход, основанный на разделении приложения на функциональные срезы и уровни, что упрощает масштабирование и сопровождение.
- **Container/Presentational Components** — паттерн, разделяющий компоненты на логические и визуальные.

Модуль аутентификации и авторизации

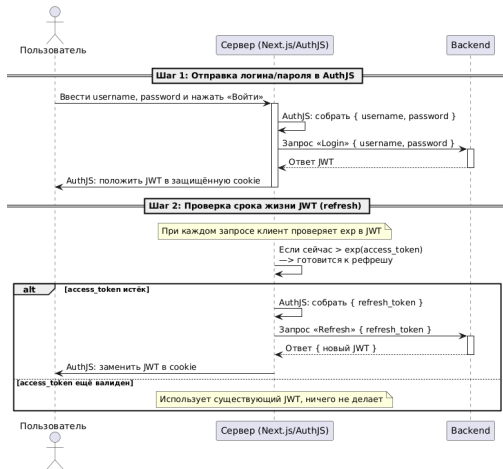


Диаграмма авторизации и аутентификации



Вход

Email *

besmilev@mail.ru

Пароль *

.....

Войти

Интерфейс входа

Взаимодействие администратора с web-приложением

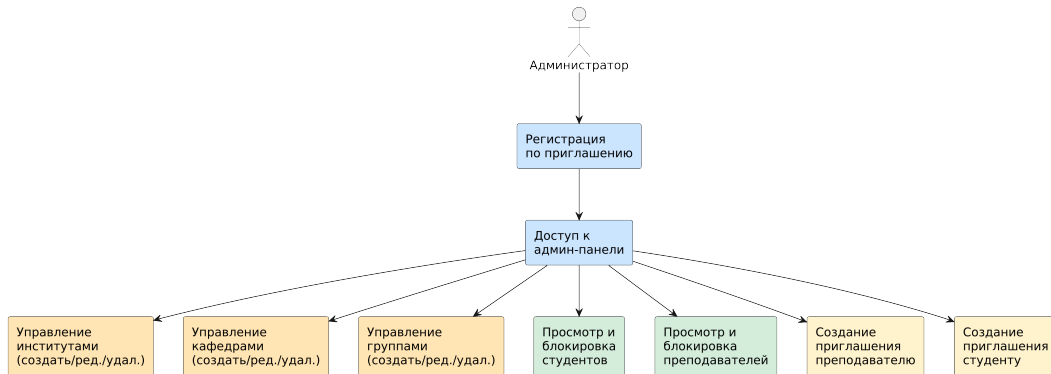


Диаграмма действий администратора

Взаимодействие студента и преподавателя с web-приложением

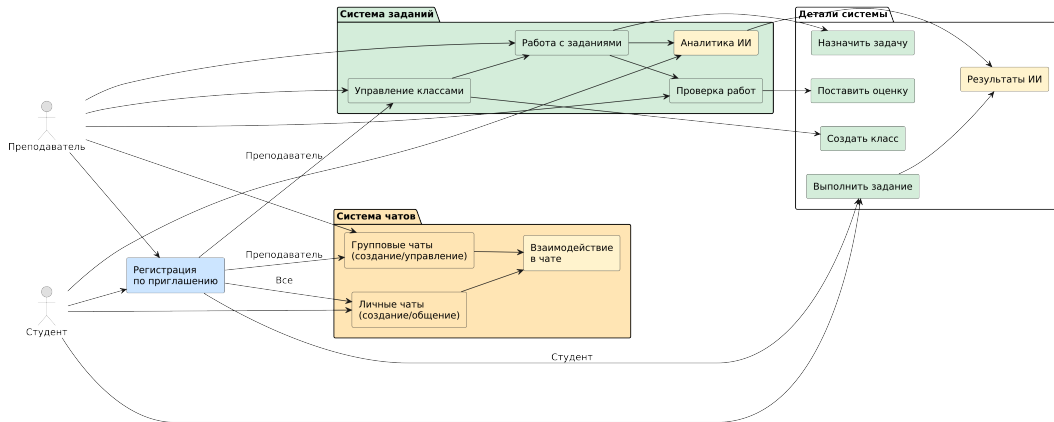


Диаграмма действий студента и преподавателя

Создание задачи

[← Назад](#)

Создать задачу

Название

Описание

Первое задание посвящено ознакомлению со средой разработки CLINE.

Прикрепить

Файл

Добавить

Задание.pdf
244 KB 256

Крайний срок сдачи

Количество баллов

10 ^ v

Анализ лабораторной работы с помощью ИИ

Запрос для анализа

Проверить файл на выполнение задания.

Интерфейс создания задачи

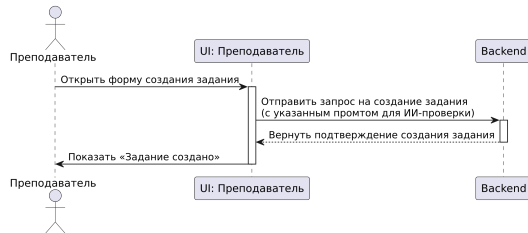


Диаграмма создания задачи

Отправка задания на проверку

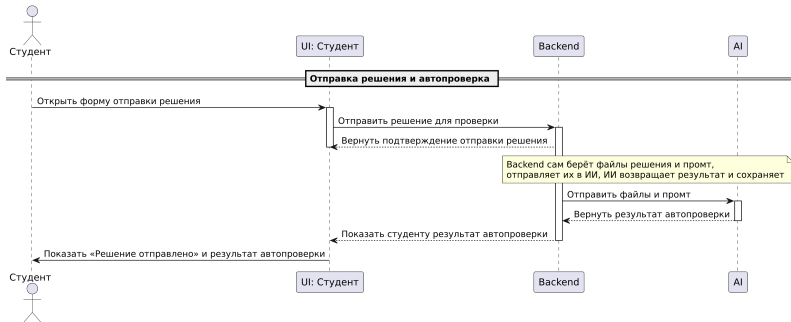
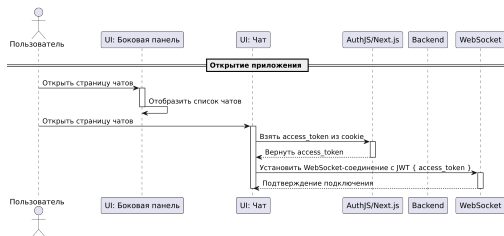
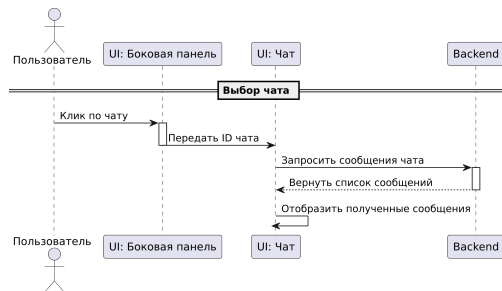


Диаграмма взаимодействия при просмотре задания конкретного студента

Диаграммы взаимодействия с чатами

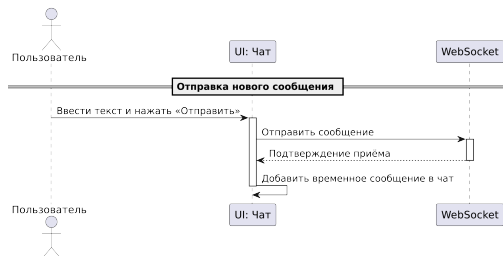


Подключение и инициализация чата



Открытие и отображение списка чатов

Диаграммы отправки и получения сообщений



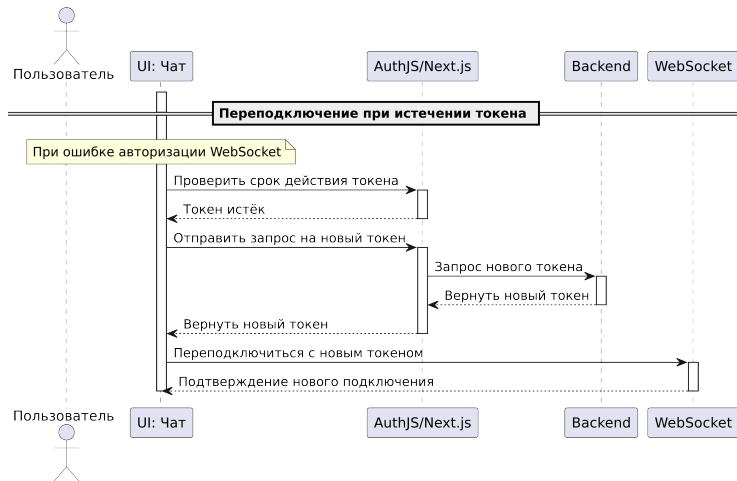
Отправка сообщения пользователем



Получение сообщения получателем

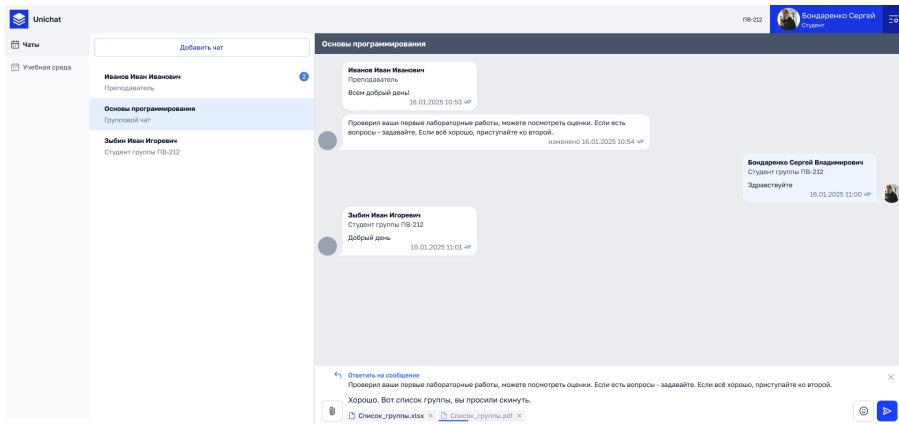
Переподключение WebSocket: диаграмма последовательности

При истечении срока действия access-токена реализован механизм его автоматического обновления и повторного переподключения WebSocket-соединения.



Интерфейс чата

Пользовательский интерфейс чата предоставляет возможность обмена сообщениями в режиме реального времени.



Тестирование

В ходе работы были протестированы следующие модули:

- **Web-приложение:**

- Модуль чатов
- Модуль редактора текста

- **UI-библиотека:**

- Модуль создания форм
- Модуль пикера даты и времени

[illegible]

Скриншот результатов тестирования Web-приложения

[illegible]

Скриншот результатов тестирования UI-библиотеки

Заключение

В ходе выполнения выпускной квалификационной работы поставленные задачи были успешно решены. Результатом стало веб-приложение с системой управления заданиями, поддерживающей автоматическую проверку решений студентов с помощью ИИ, а также чатами для общения в реальном времени через WebSocket.