

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 образовательной онлайн-платформы с возможностью
автоматизированной проверки задач по программированию**

Автор работы: Бондаренко Сергей Владимирович

Направление подготовки: 09.03.04 «Программная инженерия»

Руководитель: Мельников Антон Борисович, Директор департамента автоматизации
бизнеса

Цель и задачи

Цель: улучшить и упростить опыт преподавателей и студентов через разработку клиентской части единого образовательного приложения с обменом сообщениями, автоматической проверкой решений и удобным интерфейсом.

Задачи:

- 1 Провести анализ существующих образовательных решений.
- 2 Определить подходящую архитектуру и технологический стек.
- 3 Разработать пользовательский интерфейс, ориентированный на преподавателей и студентов.
- 4 Реализовать модули управления учебными структурами, заданиями и системой общения.
- 5 Интегрировать модуль ИИ-проверки кода.
- 6 Выполнить 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.

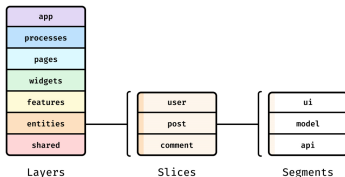
Из-за разделения функций между разными сервисами преподаватели и студенты вынуждены постоянно переключаться между несколькими приложениями, что создаёт неудобства и снижает эффективность работы.

Функция	Управление	Общение	AI-анализ
Google Classroom	+	—	—
MS Teams	+	+	—
Moodle	+	—	—
VK	—	+	—
Telegram	—	+	—
CodeSignal	—	—	+
Codility	—	—	+

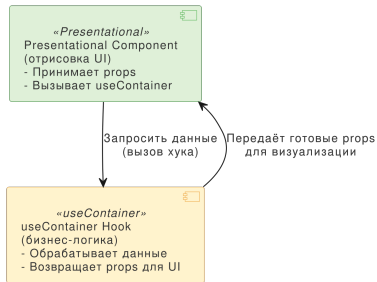


- **Next.js** — используется для серверного рендеринга (SSR), маршрутизации и повышения производительности клиентского приложения.
- **React** — обеспечивает декларативный подход к построению компонентного пользовательского интерфейса.
- **TypeScript** — добавляет статическую типизацию, улучшает читаемость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

Архитектура 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

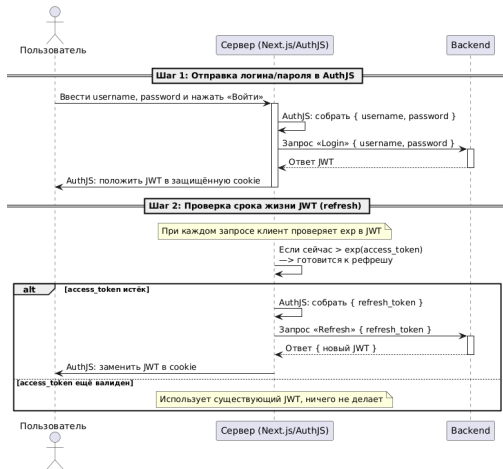


Диаграмма авторизации

Unichat

Вход

Email *

besmillev@mail.ru

Пароль *

Войти

Интерфейс входа

Административная панель

Пригласить студентов
Сформировать ссылку, по которой студенты могут пройти тест

Пригласить студентов

Email:

Исступ:

Кафедр:

Группа:

Ссылка:

Сформировать ссылку

Форма приглашения студента

Пригласить преподавателя
Сформировать ссылку, по которой студенты могут пройти тест

Пригласить преподавателя

Email:

Исступ:

Кафедр:

Ссылка:

Сформировать ссылку

Форма приглашения преподавателя

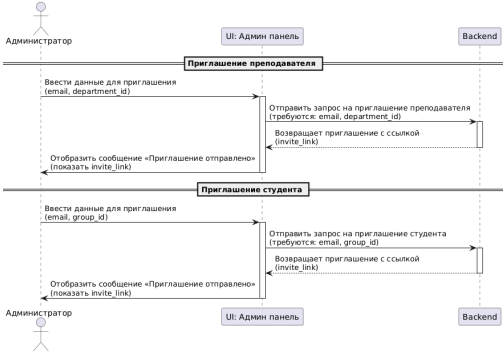



Диаграмма приглашения пользователей

Регистрация

 **Университет**

Регистрация

Институт: ИТУС
Кафедра: ПОБТАС

Ваше имя *

Пол *

Фамилия *

аds@bail.ru

Email *


soand@gmail.com

Пароль *

Подтвердите пароль *

[Заполнить форму](#)

Регистрация преподавателя

 **Университет**

Регистрация

Институт: ИТУС
Кафедра: ПОБТАС
Группа: ПБ-212

Имя *

Пол *

Фамилия *

аds@bail.ru

Email *

soand@gmail.com

Пароль *

Подтвердите пароль *

Подтвердите пароль *

[Заполнить форму](#)

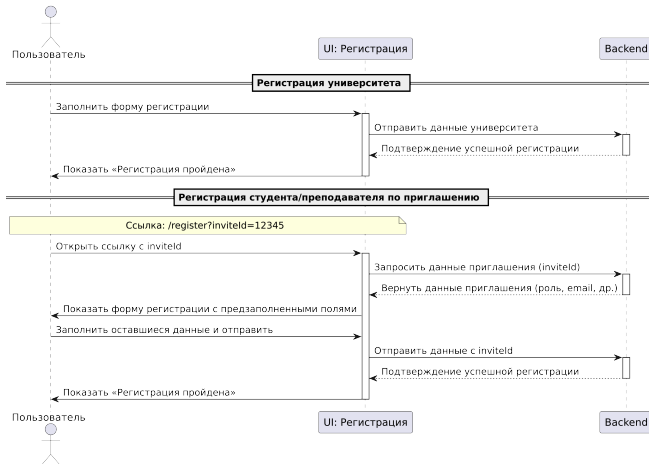


Диаграмма регистрации пользователя

Регистрация студента

Создание класса

Unicloud

Главная

Учебные курсы

Создание класса

Название

Основы Программирования

Курсы

PB-221, PB-221, BT-222, BT-221

Описание

Основы программирования - учебник для студентов, направленных на формирование базовых знаний и навыков программирования. В курсе рассматриваются ключевые понятия: переменные, типы данных, алгоритмы, условия, циклы, функции и основы работы с массивами. Особое внимание уделяется структуре данных и принципам разработки простых программ.

Материалы

Файл

Добавить

Материалы.pdf

Интерфейс создания класса

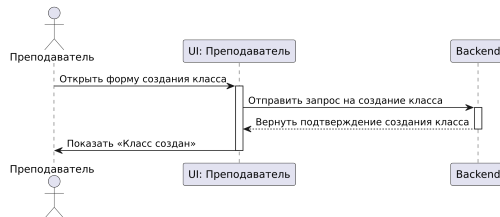


Диаграмма создания класса

Создание задачи

The screenshot shows a web interface for creating a task. The form is titled 'Создать задачу' and includes the following fields and sections:

- Название:** A text input field containing 'Лабораторная работа 1'.
- Описание:** A text area containing 'Первое задание по созданию комплексного сайта с использованием Django'.
- Прикрепить:** A section with a 'Файл' input field and a 'Добавить' button. Below it, there is a checkbox for 'Загрузка PDF' and a red 'X' icon.
- Крайний срок сдачи:** A date picker field showing '25.01.25'.
- Количество баллов:** A numeric input field showing '10'.
- Additional information:** A section with a title 'Анализ лабораторной работы с помощью ИИ' and a sub-section 'Вопрос для анализа' containing the text 'Проверить файл на выполненное задание'.

Интерфейс создания задачи

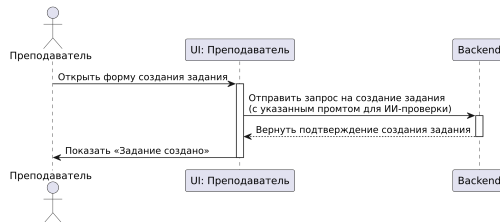
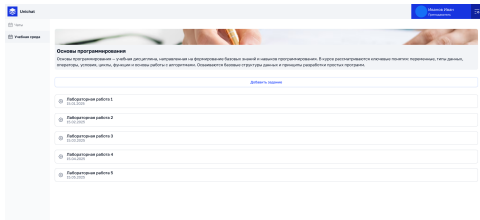
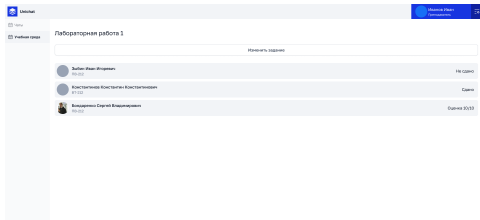


Диаграмма создания задачи

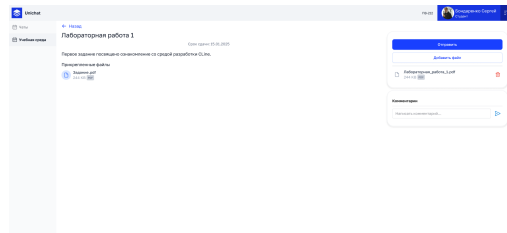
Список и просмотр заданий



Список заданий



Просмотр задания преподавателем



Просмотр задания студентом

Отправка задания на проверку

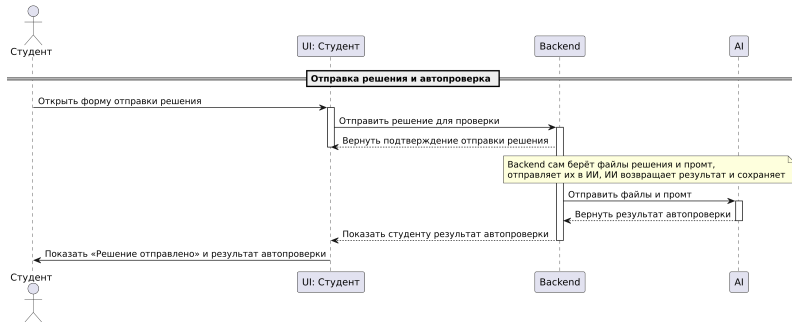
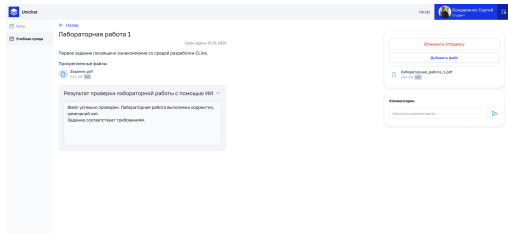
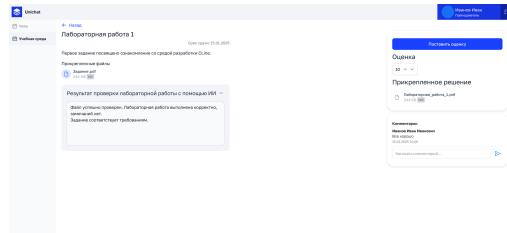


Диаграмма взаимодействия при просмотре задания конкретного студента

Интерфейс задания определённого студента

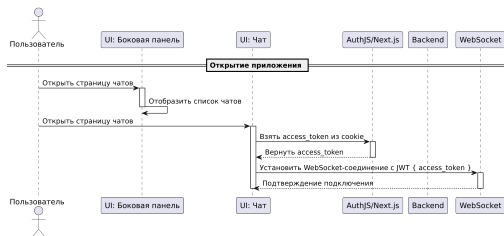


Интерфейс задания со стороны студента

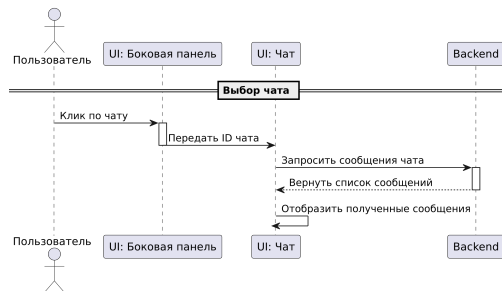


Просмотр задания преподавателем

Диаграммы взаимодействия с чатами

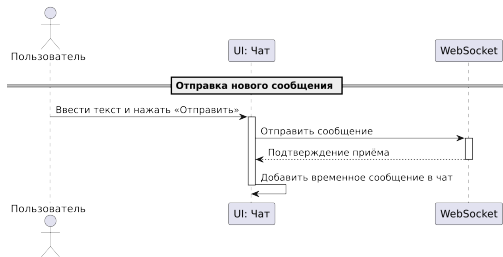


Подключение и инициализация чата

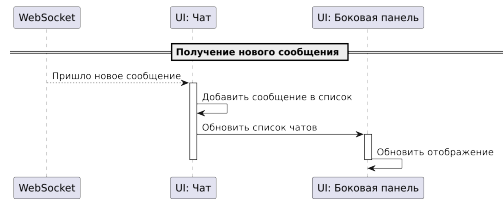


Открытие и отображение списка чатов

Диаграммы отправки и получения сообщений

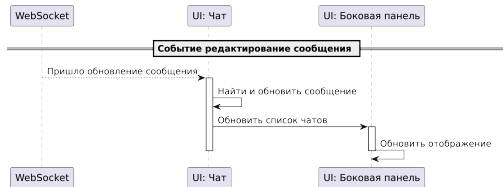


Отправка сообщения пользователем



Получение сообщения получателем

Диаграммы редактирования и чтения сообщений



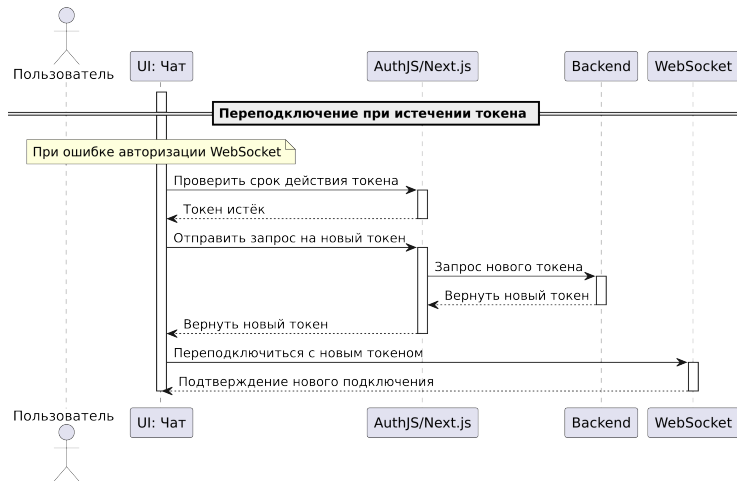
Редактирование уже отправленного сообщения



Отметка о прочтении получателем

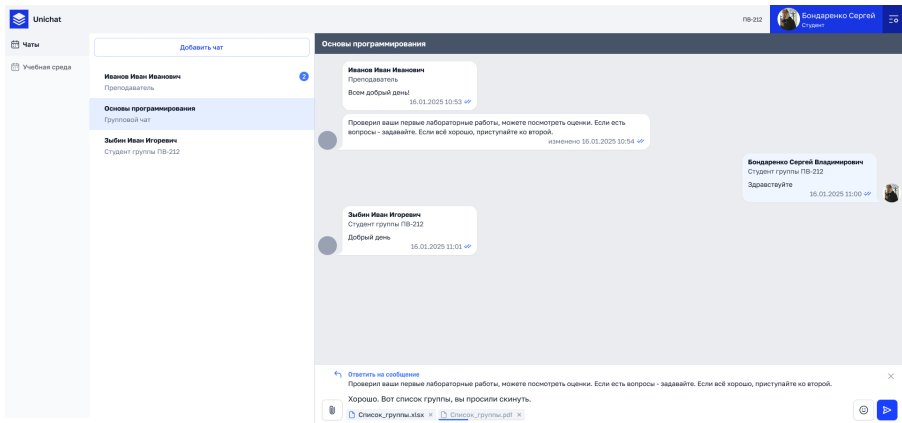
Переподключение WebSocket: диаграмма последовательности

В случае потери соединения с сервером реализован механизм автоматического переподключения. При восстановлении соединения клиент и сервер синхронизируют состояние, включая активный чат и непрочитанные сообщения.



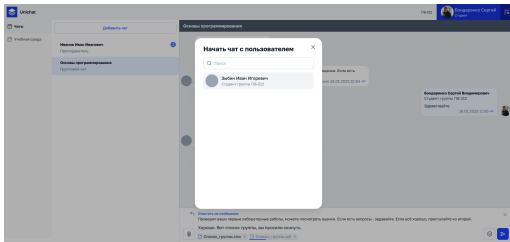
Интерфейс чата

Пользовательский интерфейс чата предоставляет возможность обмена сообщениями в режиме реального времени. Поддерживается отображение истории сообщений и отправка файлов.

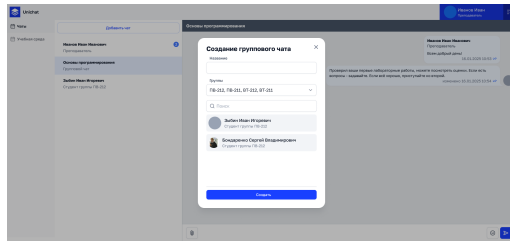


Интерфейс создания чатов

Форма создания чатов позволяет преподавателю инициировать личные и групповые чаты со студентами. Поддерживается выбор участников, автоматическое объединение по классам и создание личных чатов.



Личный чат



Групповой чат

Результаты выполнения тестов

В ходе работы была протестирована тяжёлая бизнес-логика приложения. Также проведено тестирование собственной UI-библиотеки.

```
jest
PASS src/__tests__/ExpertsChat/mergeMessages.test.ts
mergeMessages
  ✓ base перед insert (15 ms)
  ✓ base после insert
  ✓ base и insert пустые (1 ms)
  ✓ base и insert совпадают (1 ms)
  ✓ base и insert пересекаются base перед insert (1 ms)
  ✓ base и insert пересекаются base после insert
PASS src/__tests__/index.ts
Init
  ✓ Test (8 ms)
PASS src/__tests__/TextEditor/processFunction.test.ts
setValueInCode
  ✓ Пустая строка (22 ms)
  ✓ Строка без mark / Строка без node (1 ms)
  ✓ Строка с mark
  ✓ Строка с mark в конце (1 ms)
  ✓ Строка с mark в начале (1 ms)
  ✓ Строка с mark в конце и начале (1 ms)
  ✓ Строка с переходами на новую строку
  ✓ Строка с node (1 ms)
  ✓ Строка с node и большим количеством параграфов (1 ms)
  ✓ Строка обернута в blockquote и без параграфов
  ✓ Строка обернута в blockquote и большим количеством параграфов (1 ms)
  ✓ Строка с node и mark
  ✓ Строка с node и в конце \n (1 ms)
  ✓ Строка с одним \n
  ✓ Строка с множеством \n (1 ms)
Test Suites: 3 passed, 3 total
Tests: 22 passed, 22 total
Snapshots: 0 total
Time: 5.714 s
Ran all test suites.
```

Результаты в приложении

```
> jest
PASS src/__tests__/setDeepValue.test.ts (11.58 s)
setDeepValue
  ✓ Обновление значения по строковому ключу (9 ms)
  ✓ Обновление значения с использованием объекта в пути (1 ms)
  ✓ Обновление значения с комбинированным объектно-строковым путем
  ✓ Попытка обновления пустого значения (ключ отсутствует в словаре) (1 ms)
  ✓ Проверка на создание поля с нуля
  ✓ Попытка обновления пустого индексного значения (индекс отсутствует в массиве) (1 ms)
PASS src/__tests__/getDeepValue.test.ts (11.579 s)
getDeepValue
  ✓ Получение значения по строковому ключу (5 ms)
  ✓ Получение значения с использованием объекта в пути
  ✓ Получение значения с комбинированным объектно-строковым путем
  ✓ Путь к несуществующему ключу
  ✓ Попытка доступа к массиву по несуществующему индексу
  ✓ Путь с пустым объектом
PASS src/__tests__/changeInterval.test.ts (11.666 s)
changeInterval
  ✓ Дата это начало интервала (9 ms)
  ✓ Дата это конец интервала
  ✓ Интервал пустой (1 ms)
  ✓ Дата на интервале (1 ms)
  ✓ Дата до старта интервала (1 ms)
  ✓ Дата после конца интервала
  ✓ Одна data undefined, интервал состоит из одной даты
  ✓ Две data undefined, интервал состоит из двух дат
  ✓ Интервал из 2 дат, начало равно undefined (1 ms)
  ✓ Интервал из 2 дат, конец равен undefined
  ✓ Интервал из 2 дат, начало равно undefined. Конец равен дате. (1 ms)
  ✓ Интервал из 2 дат, начало равно undefined. Начало равно дате.
  ✓ Интервал имеет одну дату. Дата больше.
  ✓ Интервал имеет одну дату. Дата меньше (1 ms)
  ✓ Интервал имеет одну дату, даты равны
  ✓ Интервал из 2 дат, начало равно undefined. Начало больше даты.
  ✓ Интервал из 2 дат, начало равно undefined. Начало меньше даты. (1 ms)
  ✓ Интервал из 2 дат, конец равен undefined. Конец меньше даты.
  ✓ Интервал из 2 дат, конец равен undefined. Конец больше даты.
Test Suites: 3 passed, 3 total
Tests: 31 passed, 31 total
Snapshots: 0 total
Time: 12.448 s
Ran all test suites.
```

Результаты в UI-библиотеке

Заключение

В ходе выполнения выпускной квалификационной работы были успешно решены поставленные задачи:

- Проведён анализ существующих образовательных приложений и решений для автоматизированной проверки кода;
- Разработана архитектура клиентской части приложения с применением Feature-Sliced Design и современных паттернов проектирования;
- Разработан интуитивно понятный пользовательский интерфейс для студентов и преподавателей;
- Реализованы модули аутентификации, управления заданиями, чатами, а также внедрён механизм обмена сообщениями в реальном времени через WebSocket с обработкой переподключения и синхронизацией состояния;
- Реализована интеграция с AI для автоматизированной проверки решений;
- Проведено тестирование бизнес-логики приложения, подтверждающее корректность работы основных функций.