Санкт-Петербургский государственный университет
Кафедра системного программирования

Разработка веб-приложения для шахматной школы

Габдрахманов Азат Райнурович

Научный руководитель: к.ф.-м.н. Д.В. Луцив, доцент кафедры системного программирования

Рецензент: Д.А. Лукашук, старший программист «Таймвеб»

Санкт-Петербург
2025

Введение

- Шахматные школы сталкиваются с проблемой разрозненности учебных материалов и отсутствием инструментов аналитики для тренеров
- Существующие платформы (Chess.com, Lichess, ChessKid) ориентированы на игроков, не решают проблемы тренеров
- Проект Chess Dream создает единую платформу для шахматной школы с фокусом на потребностях тренеров
- Автоматизация рутинных задач и аналитика упрощают работу с большим количеством учеников

Постановка задачи

Цель работы: Разработка веб-приложения для поддержки тренировочного процесса шахматистов, обеспечивающего взаимодействие между тренерами и учениками

Задачи:

- Провести обзор существующих решений для обучения шахматам и выявить их ограничения
- Сформировать требования для приложения
- Спроектировать архитектуру приложения Chess Dream
- Разработать интерфейс тренера с возможностями создания и назначения персонализированных домашних заданий
- Создать интерфейс ученика для решения назначенных задач
- Разработать интерфейс для анализа результатов и прогресса учеников
- Провести апробацию приложения Chess Dream с участием шахматных тренеров и их учеников

Обзор существующих решений

Chess.com

- Огромное сообщество (100+ млн)
- Фокус на игровой процесс
- Турниры и соревнования
- В РФ не работает без VPN

Lichess

- Открытый исходный код
- Большая база задач (1+ млн)
- Ограниченные возможности анализа результатов
- Создание студий вручную

ChessKid

- Адаптирован для детей
- Игровые элементы
- Ограниченная база задач в бесплатной версии
- Нет персонализированных заданий

StepChess

- Создание домашних заданий
- Отслеживание времени и попыток
- Нельзя видеть все попытки ученика
- Нет массового импорта задач

Требования: автоматизация рутинной работы тренеров и создание платформы для шахматной школы. Домашние задания, статистика, прогресс

Требования к системе

Функциональные требования:

- Создание домашних заданий по тематике и рейтингу
- Автоматический подбор задач без повторений
- Личный кабинет ученика с историей
- Сохранение всех ходов и времени решения
- Анализ попыток с воспроизведением ходов
- Авторизация через Lichess

Нефункциональные требования:

- Поддержка 100+ одновременных пользователей
- Совместимость с Chrome, Firefox, Safari
- Простой и понятный интерфейс для детей
- Надежное сохранение данных
- Регулярное резервное копирование

Пользовательские сценарии:

- Работа в шахматном клубе
- Домашние задания и онлайн-анализ
- Предтурнирная подготовка

Архитектура приложения

Технологии:

- Frontend: React
- Backend: ASP.NET Core
- СУБД: PostgreSQL
- ORM: Entity Framework Core
- Авторизация: OAuth2 (Lichess)
- CI/CD: GitLab CI
- Мониторинг: Prometheus, Grafana

Системная организация:

- Многослойная клиент-серверная архитектура
- REST API
- Интеграция с Lichess для авторизации
- База данных шахматных задач
- Контейнеризация с Docker

Структура базы данных

Основные сущности системы:

- **puzzle** — задачи (FEN, рейтинг, тема, решение)
- **homework** — домашние задания (название, владелец, параметры)
- **homeworkpuzzle** — связь заданий и задач с порядком
- **solvedpuzzle** — успешные решения с метриками времени и попыток
- **attemptpuzzle** — детализация каждого хода ученика

Ключевые особенности:

- Сохранение полной истории действий ученика
- Возможность воспроизведения ходов ученика

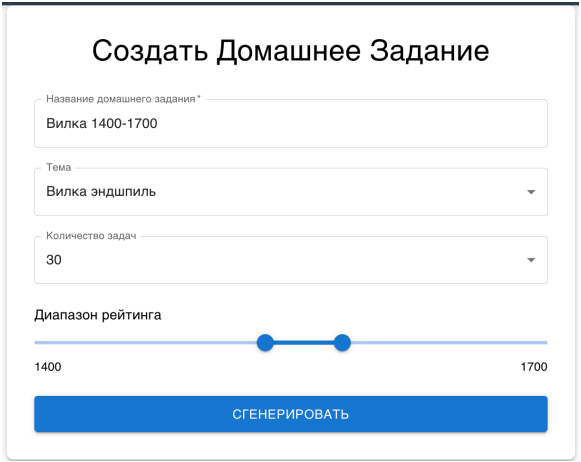
Реализация интерфейса тренера

Алгоритм подбора задач:

1. Выбор задач с рейтингом в интервале [minElo, maxElo]
2. Исключение уже назначенных тренером задач
3. Псевдослучайная выборка из оставшегося набора

Система фильтрации:

- По тематическим группам (вилка эндшпиль, связка и др.)
- По диапазону сложности (рейтинг Elo)
- Настраиваемое количество задач



Интерфейс создания домашнего задания

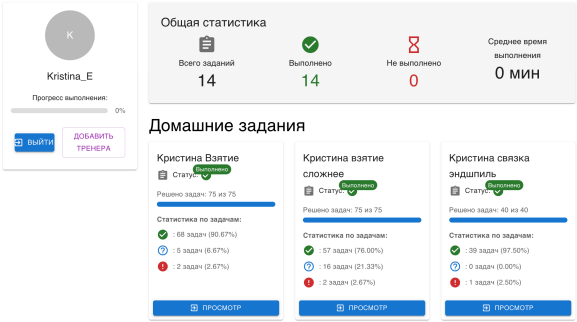
Реализация интерфейса ученика

Интерфейс решения задач:

- Интерактивная шахматная доска на React
- Сохранение каждого хода с временными метками
- обратная связь о правильности хода

Личный кабинет:

- Список назначенных заданий с прогрессом
- История выполненных задач



Личный кабинет ученика

Система аналитики и мониторинга

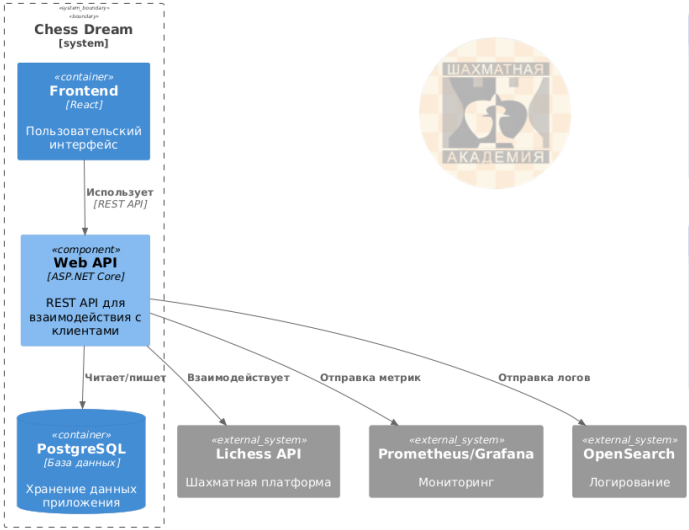
Сбор данных в реальном времени:

- **Запись каждого хода:** фиксация хода, порядкового номера и правильности
- **Отслеживание времени:** время решения задачи
- **Учёт решённых задач:** запись в solvedpuzzle с данными о времени и попытках

Аналитические возможности для тренера:

- Детальный просмотр всех попыток ученика
- Статистика по времени решения и количеству ошибок

Развертывание и инфраструктура



Основные компоненты

- Frontend на React
- Backend на ASP.NET Core
- PostgreSQL (4+ млн задач)

Мониторинг и логирование

- Prometheus/Grafana для метрик
- Serilog + OpenSearch для логов
- Автоматические алерты

Апробация

Участники исследования:

- Основной заказчик: шахматная школа UfaChess
- Тренеры UfaChess: 10 человек
- Независимые частные тренеры: 7 человек
- Ученики: 15 человек
- Родители: 7 человек

Результаты SUS (System Usability Scale):

- Тренеры UfaChess: 78.5 баллов из 100
- Независимые частные тренеры: 82.3 баллов из 100
- Средняя оценка функциональных критериев: 4.4 из 5

Метрики использования:

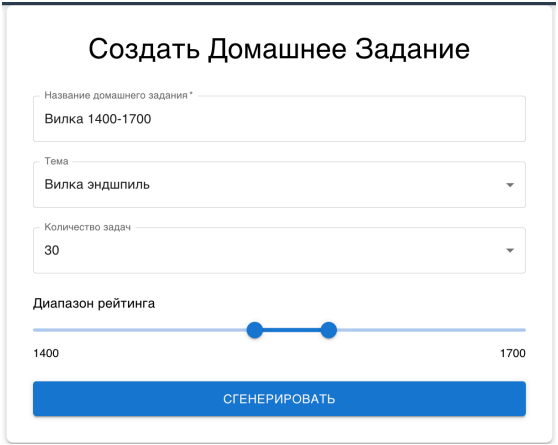
- Зарегистрировано пользователей: 150 человек
- Количество решённых задач: 15400 задач
- Общее число сделанных ходов: 44300 попыток

Результаты, выносимые на защиту

1. Выполнен обзор существующих решений с анализом их ограничений
2. Сформированы требования к приложению на основе обсуждения с тренерами шахматных школ
3. Спроектирована архитектура приложения
4. Реализован интерфейс тренера: создание и автоматическая генерация домашних заданий
5. Реализован интерфейс ученика: личный кабинет, решение задач с фиксацией ходов и времени
6. Реализован интерфейс для анализа результатов и прогресса учеников
7. Проведена апробация платформы среди основных групп пользователей

Ссылка на сайт: https://chessdream.ru

Интерфейс приложения



Интерфейс генерации задания

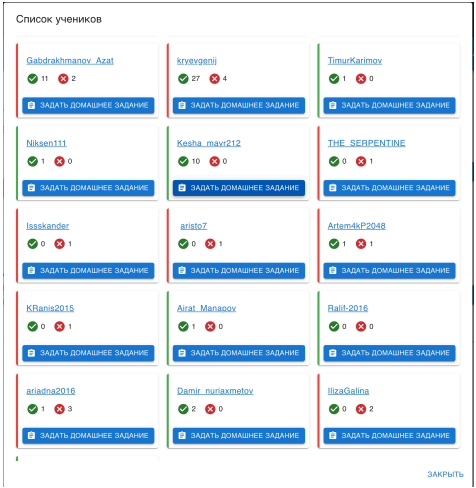
Результат

Аналитика для тренера

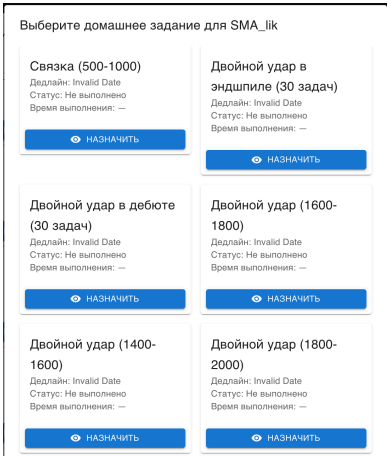
Статистика выполнения заданий

Анализ ходов ученика

Кабинет тренера



Ученики тренера



Задать домашнее задание ученику

Планы развития

Дальнейшие направления:

- Расширение геймификации и рейтинговой системы учеников
- Интеграция модуля загрузки и анализа PGN-файлов партий
- Разработка сервиса уведомлений для родителей и тренеров
- Добавление возможности задавать таймер для решения задач
- Создание мобильной версии приложения