|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Шашки Онлайн\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Студент \_\_\_\_\_ИУ5-64Б\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Н.А. Гордеев\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Студент \_\_\_\_\_ИУ5-64Б\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Я.С. Стельмах\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Студент \_\_\_\_\_ИУ5-64Б\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_А.Д. Федотова\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_В.А. Галкин\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2023 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(И.О.Фамилия)

« \_\_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине \_\_\_\_\_\_\_\_\_Сетевые технологии в АСОИУ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студенты группы \_\_\_\_ИУ5-64Б\_\_\_\_\_\_

\_\_\_\_Гордеев Никита Андреевич, Стельмах Яна Сергеевна, Федотова Анастасия Дмитриевна\_

(Фамилия, имя, отчество)

Тема курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_Шашки Онлайн\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

График выполнения работы: 25% к \_\_\_ нед., 50% к \_\_\_ нед., 75% к \_\_ нед., 100% к \_\_\_ нед.

***Задание*** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Разработать распределенную систему \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_для игры «Шашки Онлайн»\_\_\_\_\_\_\_\_\_\_\_­­­\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Оформление курсовой работы:***

Расчетно-пояснительная записка на \_22\_\_ листах формата А4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания « \_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_В.А. Галкин\_\_**

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Н.А. Гордеев\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Я.С. Стельмах\_\_\_**

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_А.Д. Федотова**\_**\_**\_

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

[Введение 4](#_gjdgxs)

[Назначение программного продукта 6](#_30j0zll)

[Архитектура программного продукта 8](#_1fob9te)

[Сервис бэкенда 10](#_3znysh7)

[Фронтенд 13](#_2et92p0)

[Сервис WebSocket 20](#_tyjcwt)

[Заключение 22](#_3dy6vkm)

[Список использованных источников 23](#_1t3h5sf)

[Приложения 24](#_4d34og8)

# Введение

Основной целью курсовой работы являлось создание распределенной системы на примере игры «Шашки Онлайн». Компоненты онлайн-игры взаимодействуют между собой различными способами: при помощи HTTP запросов, websocket и gRPC.

Распределенная система «Шашки Онлайн» предназначена для онлайн игры в шашки.

В функционал данного приложения входит:

* Просмотр в Личном кабинете пользователя информации о всех пройденных играх, количестве заработанных очков, общем количестве сыгранных игр, количестве побед, а также рейтинга пользователя.
* Авторизация/регистрация на сайте. Посетитель может авторизоваться в роли игрока. В зависимости от того, авторизован ли пользователь, ему будет доступен разный функционал.
* Возможность поиска игры, а также создание своей собственной.
* Пользователь может ознакомиться с подробными правилами игры в шашки.
* Возможность фактической игры в игру «Шашки Онлайн». Пользователь может выиграть противника, проиграть ему, а также возможна ничья.
* Просмотр полного рейтинга всех участников, когда-либо игравших в «Шашки Онлайн».

Мгновенное уведомление всех пользователей о том, что какой-то игрок создал запрос на поиск оппонента обеспечивается использованием протокола WebSocket, а также WebSocket обеспечивает передачу информации о сделанном ходе во время игры.

Данная работа была разбита на несколько составляющих частей:

* Фронтенд — реализовать приложение на React + Redux Toolkit + Axios + MUI. Необходимо реализовать окно регистрации и авторизации. Приложение должно общаться как с веб-сервисом Django-Backend, так и по веб-сервису WebSocket.
* Бэкенд — реализовать веб-сервис (API gateway), который будет предоставлять методы для фронтенда. Веб-сервис взаимодействует с базой данных. Необходимо ограничить доступ к методам сервиса через авторизацию.
* Интеграционная задача — реализовать протокол прикладного уровня для передачи по WebSocket информации об открытых играх и информации о ходах. Сервис ws взаимодействует с бэкендом по gRPC для получения/изменения данных. Гарантированная доставка сообщений при повторном подключении после разрыва соединения.

# Назначение программного продукта

Роли и функции пользователей программного продукта отражены на диаграмме прецедентов:

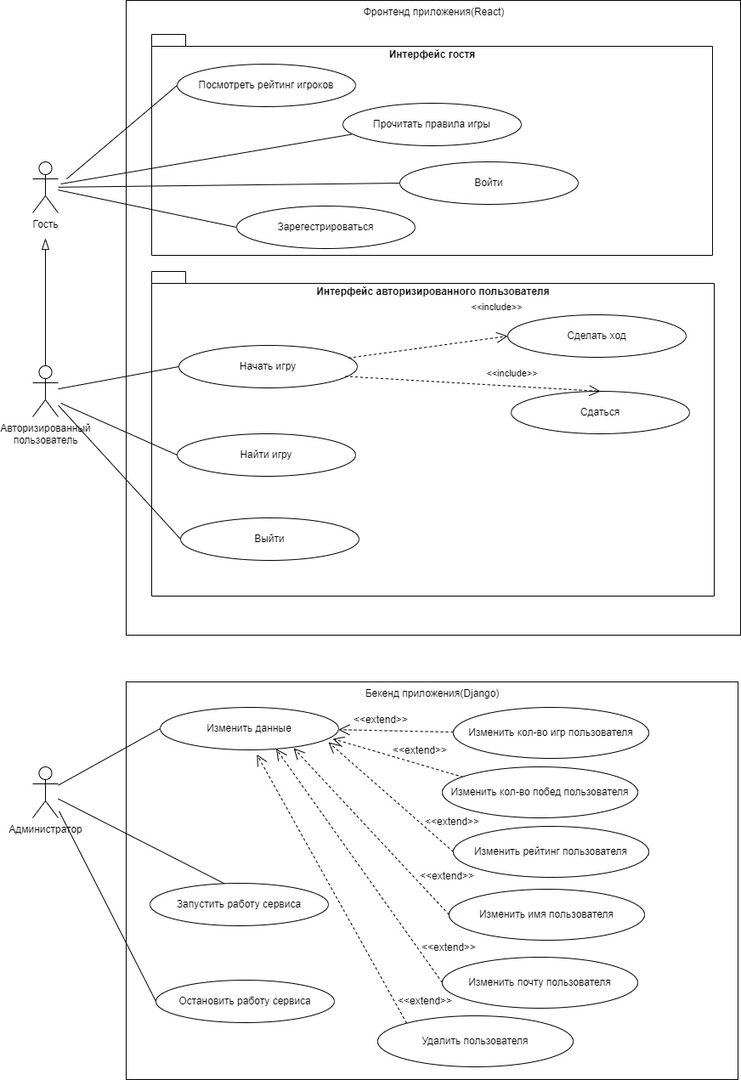


Рисунок 1 - Диаграмма прецедентов

Распределённая система:

1. Обеспечивает авторизацию и регистрацию пользователей;
2. Обеспечивает разделение пользователей на администраторов, авторизованных пользователей и гостей;
3. Обеспечивает гостю и авторизованному пользователю просмотр рейтинга игроков, а также просмотр и чтение правил игры;
4. Обеспечивает авторизованному пользователю возможность найти игру, посмотреть предыдущие игры, создать игру, сыграть в нее, а именно делать ходы, сдаться, а также выйти из аккаунта.
5. Обеспечивает интерфейс администраторов для доступа к базе данных;
6. Обеспечивает администраторам возможность запустить или остановить работу сервиса;

# Архитектура программного продукта

Архитектура программного продукта представлена на диаграмме развёртывания.

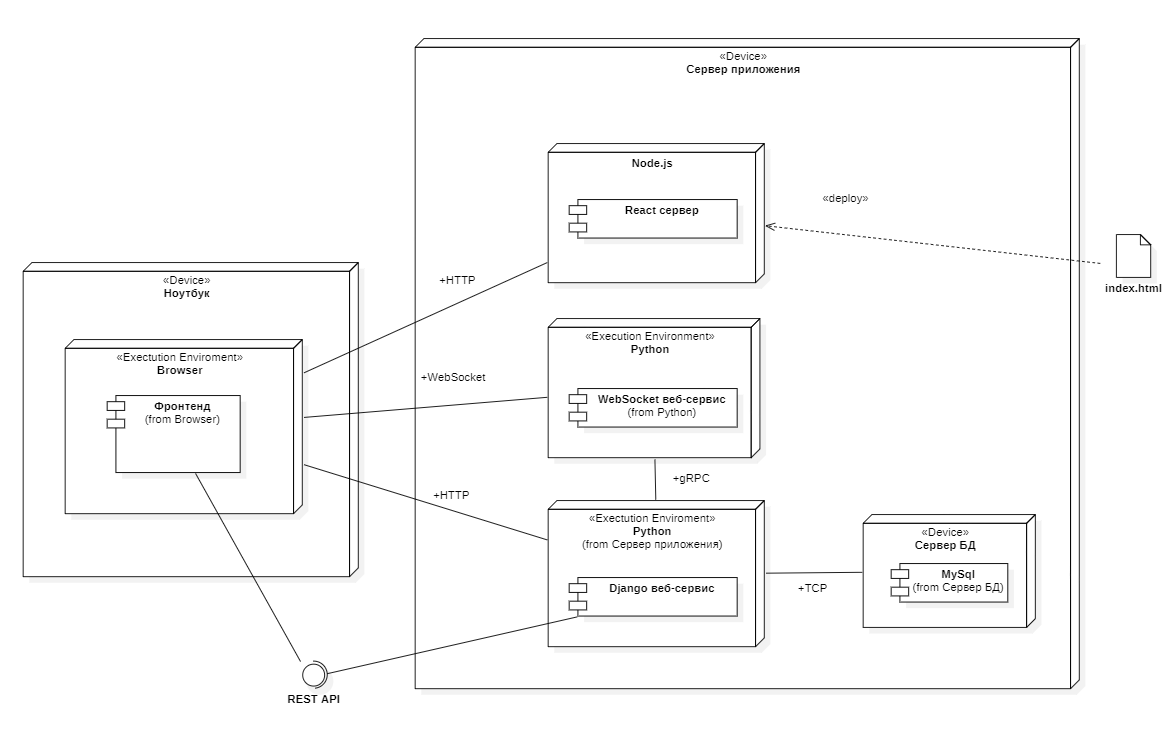


Рисунок 2 - Диаграмма развёртывания

Распределённая система состоит из бэкенда, фронтенда и сервиса Websocket. Пользователь взаимодействует с программой при помощи пользовательского интерфейса. С его помощью пользователь имеет ограниченный доступ к базе данных через бэкенд приложения. Взаимодействие бэкенда и фронтенда происходит при помощи AJAX запросов.

Протокол Websocket используется в реальном времени для обновления информации о доступных для игры играх и о совершенных игроками ходах. Для подключения клиентов ws разработан отдельный сервис. Взаимодействие фронтенда и сервиса ws происходит по протоколу Websocket. Для передачи новой информации, полученной с бэкенда, бэкенд и сервис ws взаимодействуют по gRPC.

# Сервис бэкенда

Сервис бэкенда выполнен на языке Python с использованием Django Rest Framework. Взаимодействие с фронтендом происходит по Rest API, взаимодействие Websocket и Django-Backend происходит по gRPC. Сервис бэкенда используется для удалённого вызова процедур для обращения к базе данных.

Бэкенд взаимодействует с сервером базы данных. В качестве СУБД используется MySQL. Схема базы данных представлена на ER диаграмме.

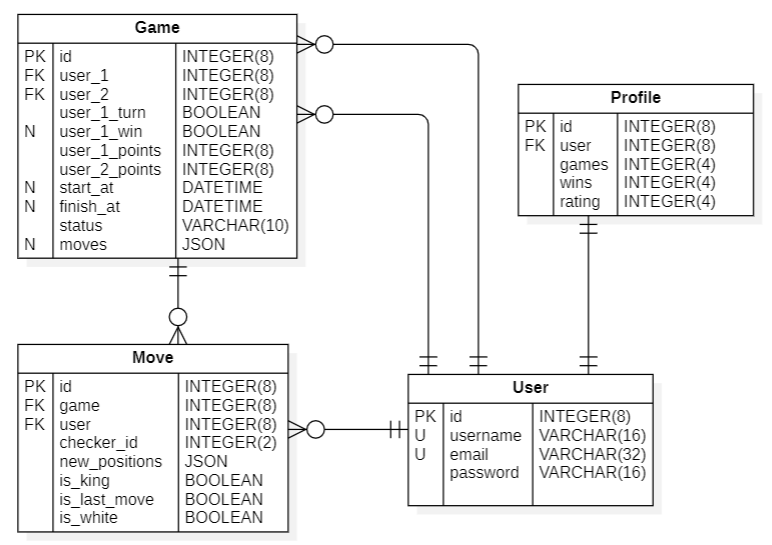


Рисунок 3 - ER диаграмма

База данных позволяет хранить все необходимое:

* логин и пароль пользователя, его почту;
* информацию о личном кабинете игрока с логином пользователя, количеством сыгранных игр, количеством побед и рейтингом;
* данные игры: логины 2-х игроков, их очки, флаги очередности и победы, ходы игроков, статус игры, а также время ее начала и конца;
* информацию о ходе пользователя: в какой игре был сделан ход, кто его выполнил, какой пешкой, где теперь стоит пешка, проверка стала ли пешка дамкой, белая ли она, а также последнюю информацию о местоположении пешки на игровом поле.

Бэкенд предоставляет фронтенду данные из базы данных в виде JSON, а также может делать новые записи в базу данных. Сервис предоставляет методы для:

* регистрации: пользователю необходимо указать свою электронную почту, логин и 2 одинаковых пароля (для подтверждения корректного ввода), переданные данные сохраняются в базу данных;
* авторизации: пользователю необходимо указать почту и пароль, в ответ пользователю будут отправлены access и refresh токены, а также его username и почта;
* выхода из аккаунта: пользователь отправляет свой refresh-токен и получает разрешение на деавторизацию
* обновление access-токена: пользователь отправляет свой refresh-токен и получает новый access токен
* просмотра профиля пользователя: пользователю необходимо отправить пустой запрос на соответствующий URL, так как метод доступен только авторизованному пользователю, id пользователя будет взят из JWT токена, в ответ пользователю будет отправлен его портфель;
* получение ходов игры: пользователю нужно указать id игры, ходы из которой он хочет получить;
* просмотр игры: . пользователю нужно указать id игры, информацию по которой он хочет получить

Доступные методы показаны на диаграмме последовательности:

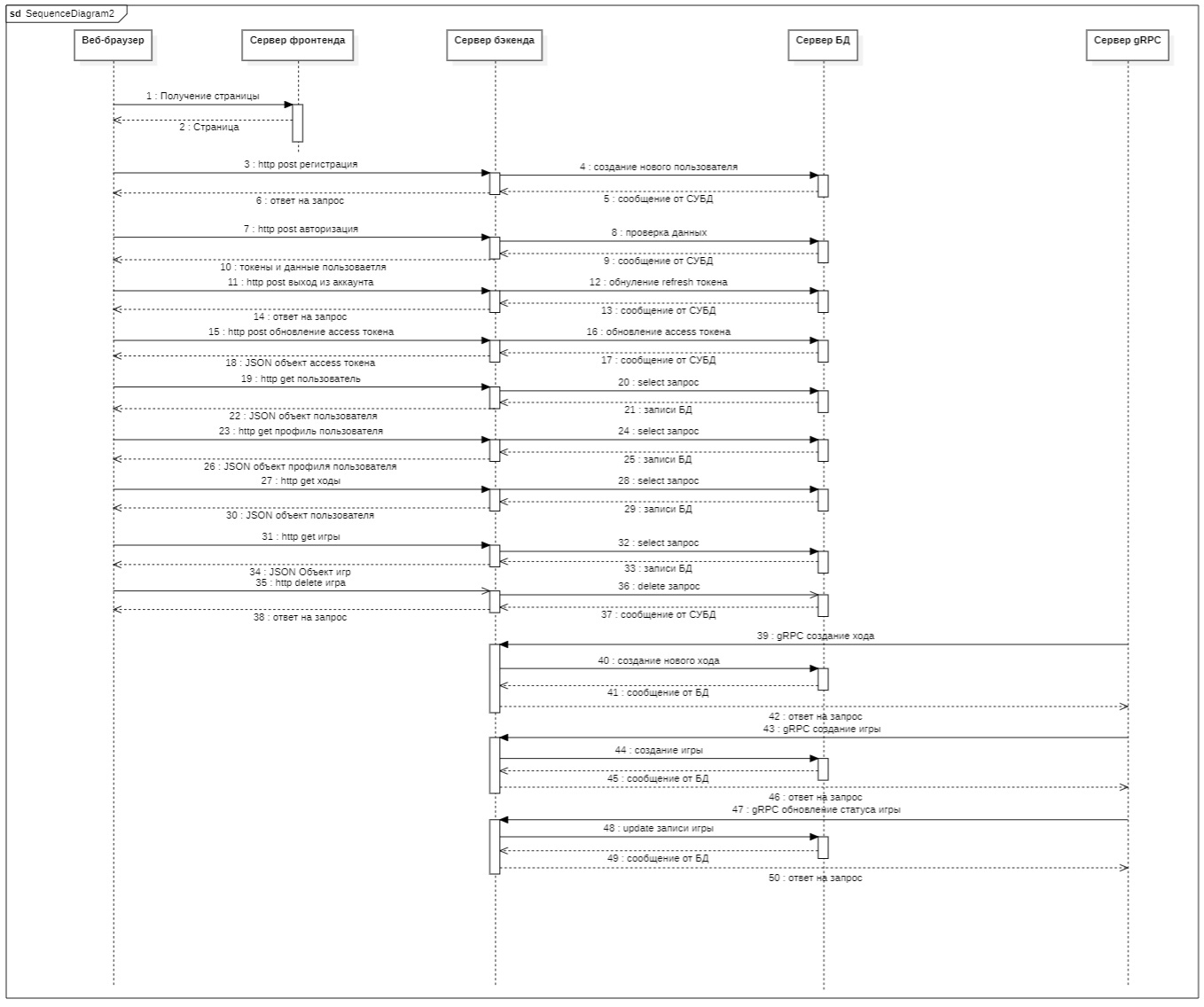


Рисунок 4 - Диаграмма последовательности бэкенда

Авторизация происходит по JWT, помимо служебной информации, в токене также хранится id пользователя, которому принадлежит токен.

К backend также обращается webSocket по gRPC. WS может использовать метода для создания и изменения записи игры или создания записи хода.

# Фронтенд

Для создания макета приложения был использован онлайн-сервис Figma. Для унификации интерфейса была использована библиотека MUI. MUI предоставляет широкий выбор компонентов, которые также значительно помогают сократить время написания приложения.

Таблица DataGrid для отображения таких данных как сыгранные пользователем игры или доступный к игре игры. Таблица имеет очень гибкий интерфейс настройки для каждого столбца отдельно, позволяя фильтровать, сортировать и выбирать данные.

Фронтенд разработан на JavaScript с учетом рекомендаций по простоте, удобству и функциональности интерфейса. Для создания внешних пользовательских интерфейсов в процессе разработки была использована библиотека React, которая является открытым исходным кодом. React отличается от других JavaScript-библиотек тем, что ориентирована на создание компонентов, которые инкапсулируют состояние и генерируют элементы пользовательского интерфейса, вместо предоставления полноценной платформы приложений.

В проекте также использовался язык программирования TypeScript, поскольку JavaScript не является типизированным языком. Использование TypeScript имеет следующие преимущества: строгая типизация значительно снижает возможность ошибок, он удобно интегрируется с IDE, расширяет возможности JavaScript в области ООП (наследование, абстрактные классы и т.д.), компилятор TypeScript позволяет преобразовывать код в более старые стандарты языка JavaScript, что обеспечивает лучшую поддержку браузерами, а также улучшает читаемость и документируемость кода.

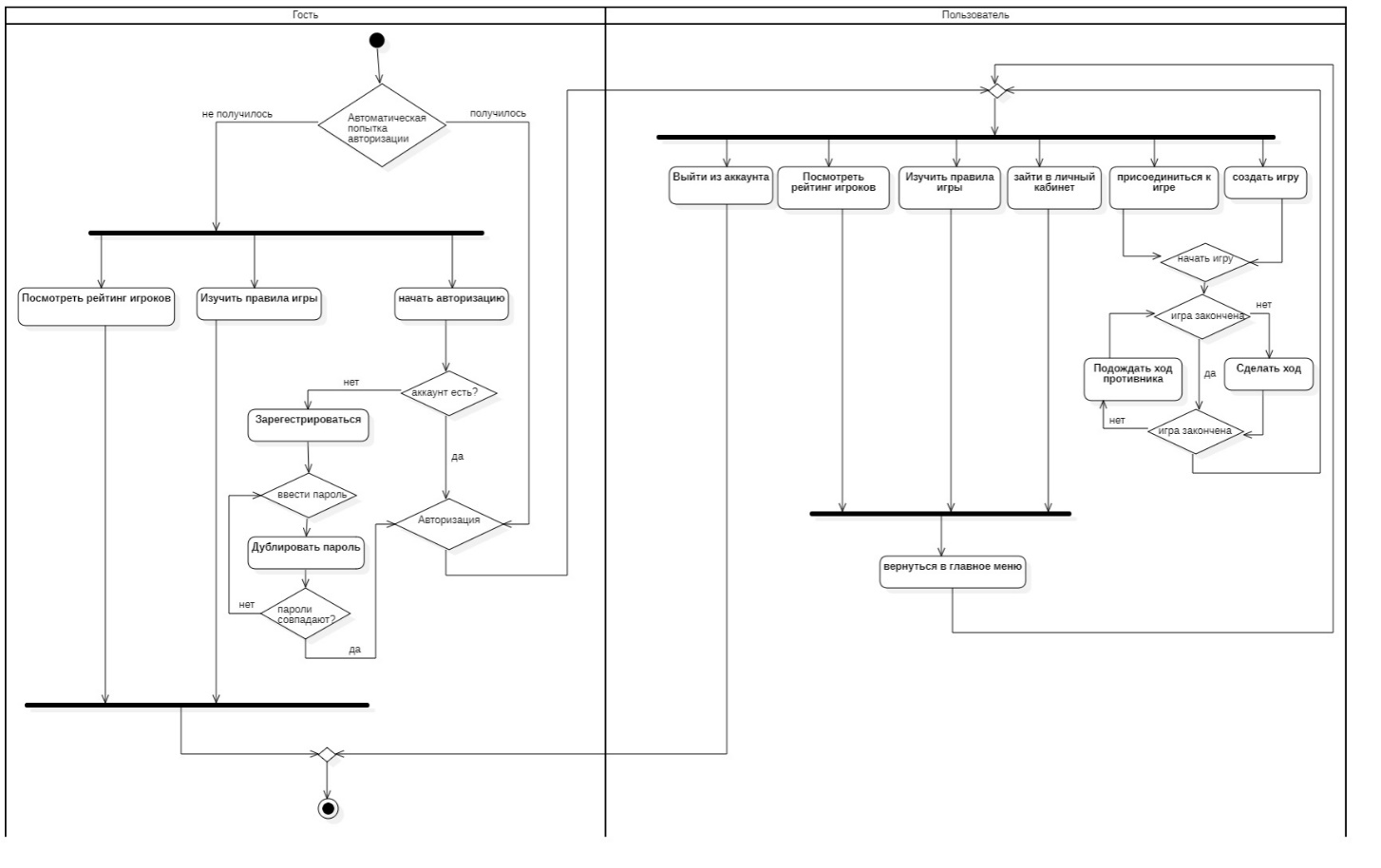
В проекте используется библиотека redux-toolkit для управления состоянием.

Для отправки запросов на бэкенд применяется библиотека Axios. Axios является легковесным клиентом HTTP, построенным на основе сервиса $http из Angular.js, и имеет схожий с ним JavaScript Fetch API. Благодаря использованию этой библиотеки, взаимодействие между фронтендом и бэкендом значительно упрощается.

Для обеспечения актуальности данных о стоимости акций применяется протокол WebSocket, который обеспечивает непрерывное соединение между сервером и клиентом, позволяя передавать новые данные в клиентскую базу данных при их обновлении.

Для реализации навигации используется библиотека react-router-dom.

Взаимодействие пользователя с приложением представлено на диаграмме деятельности.



При запуске приложения посетитель попадает на главную страницу. До авторизации он считается гостем и может просматривать рейтинг игроков, изучить правила игры в шашки или начать авторизацию. Если аккаунта до этого времени не было создано, пользователь имеет возможность совершить регистрацию. Для этого он вводит необходимые данные, придумывает и для достоверности корректного ввода дублирует пароль. Если пароли совпадут, регистрация будет считаться успешной. Гость может авторизоваться как обычный пользователь, введя свои логин и пароль в соответствующих полях. Далее данные отправляются на бэкенд для проверки корректности. В случае верной авторизации, пользователю возвращаются access и refresh токены, его username и почта, записывающиеся в Redux. Авторизованному пользователю доступны следующие функции: выход из аккаунта, просмотр рейтинга игроков, чтение правил игры, вход в личный кабинет, возможность присоединиться к игре или создать свою собственную игру. Помимо этого, есть возможность вернуться к главному меню.

При нажатии на иконку шашек в левом верхнем углу экрана можно попасть на главную страницу (рис. 5). Находясь на главном экране, пользователь может перейти в раздел правил, посмотреть рейтинг игроков и авторизоваться. При нажатии на кнопку «Правила игры» пользователь будет перенаправлен на страницу правил, а при нажатии на кнопку «Рейтинг игроков» - на страницу рейтинга игроков.

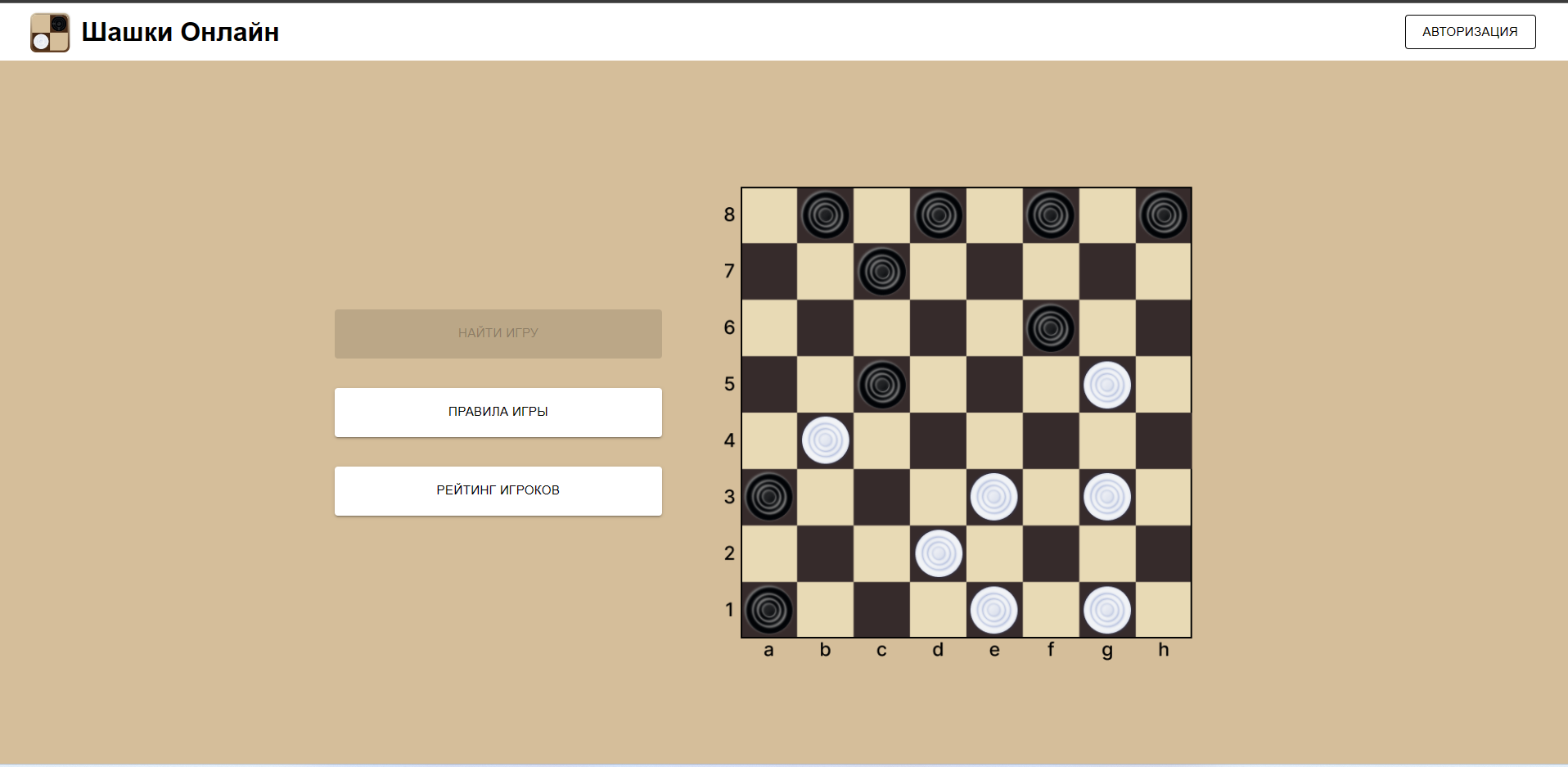


Рисунок 5 - Главный экран неавторизованного пользователя

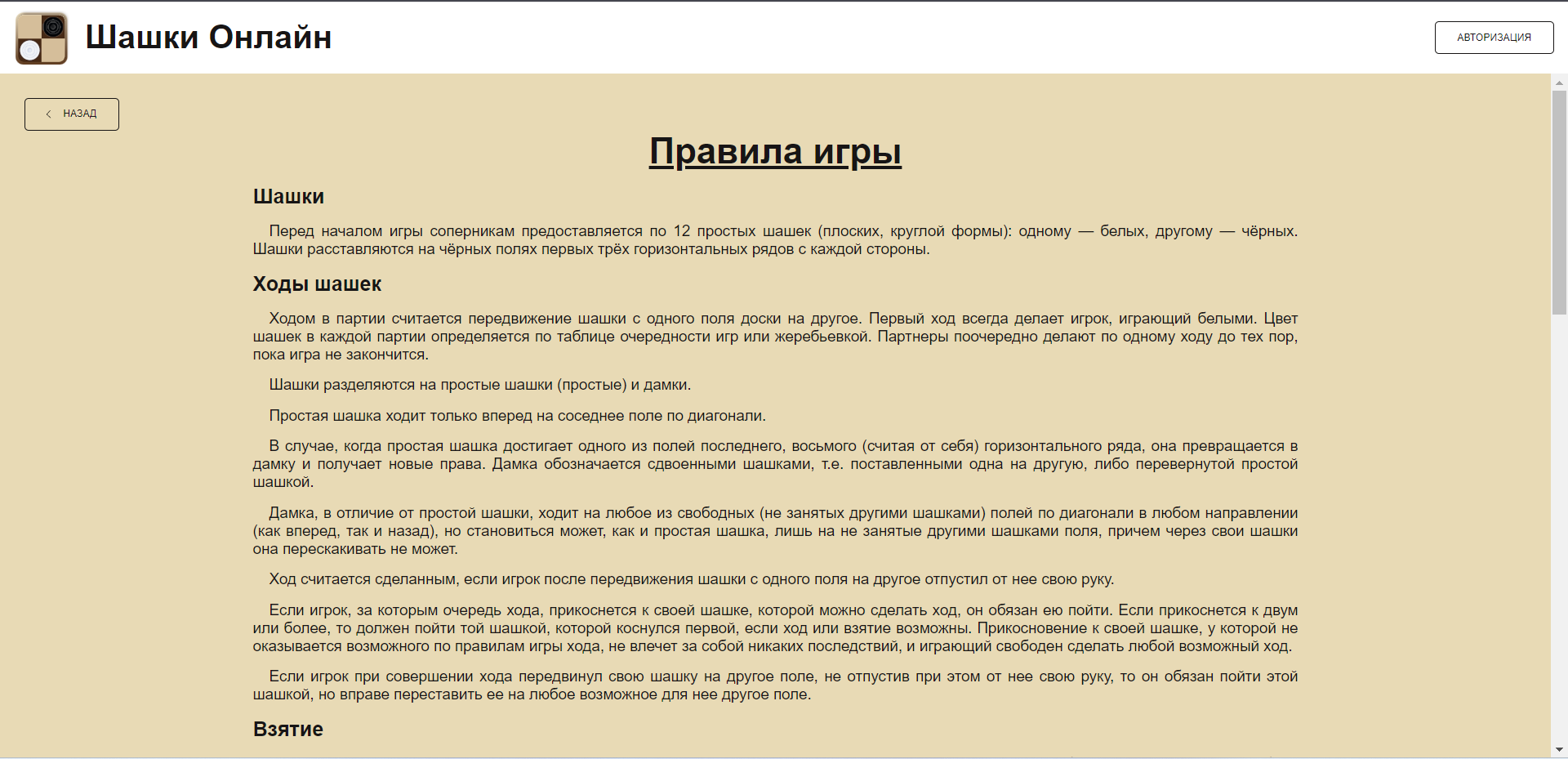


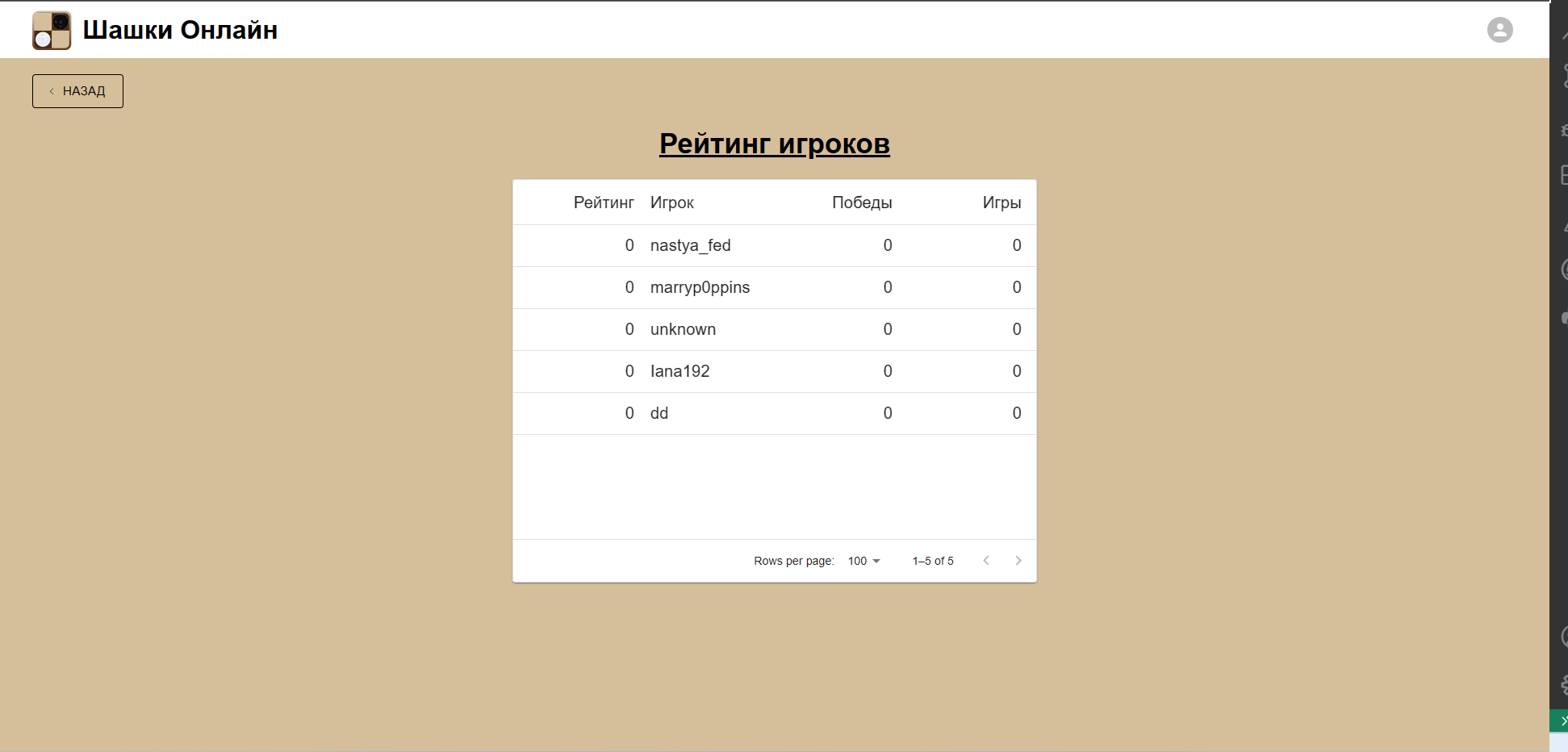
Рисунок 6 - Страница правил

Рисунок 7 - Страница рейтинга игроков

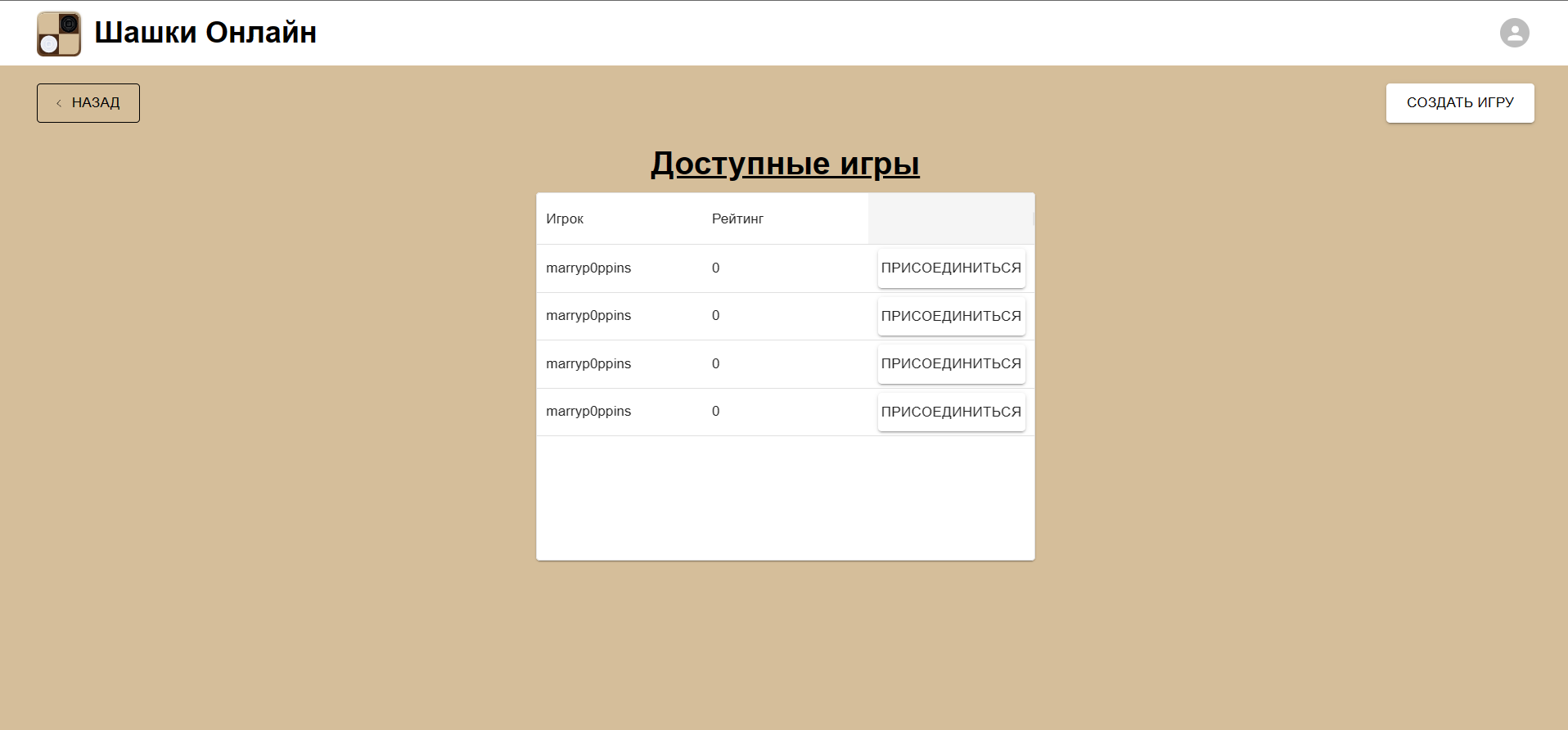
После авторизации у пользователя появляется доступ к кнопке для перехода на страницу поиска игр или создания новой игры(рис 8). 

Рисунок 8 – Меню поиска игр авторизованного пользователя

При создании новой игры или присоединения к существующей пользователя будет перенаправлять на страницу игры.

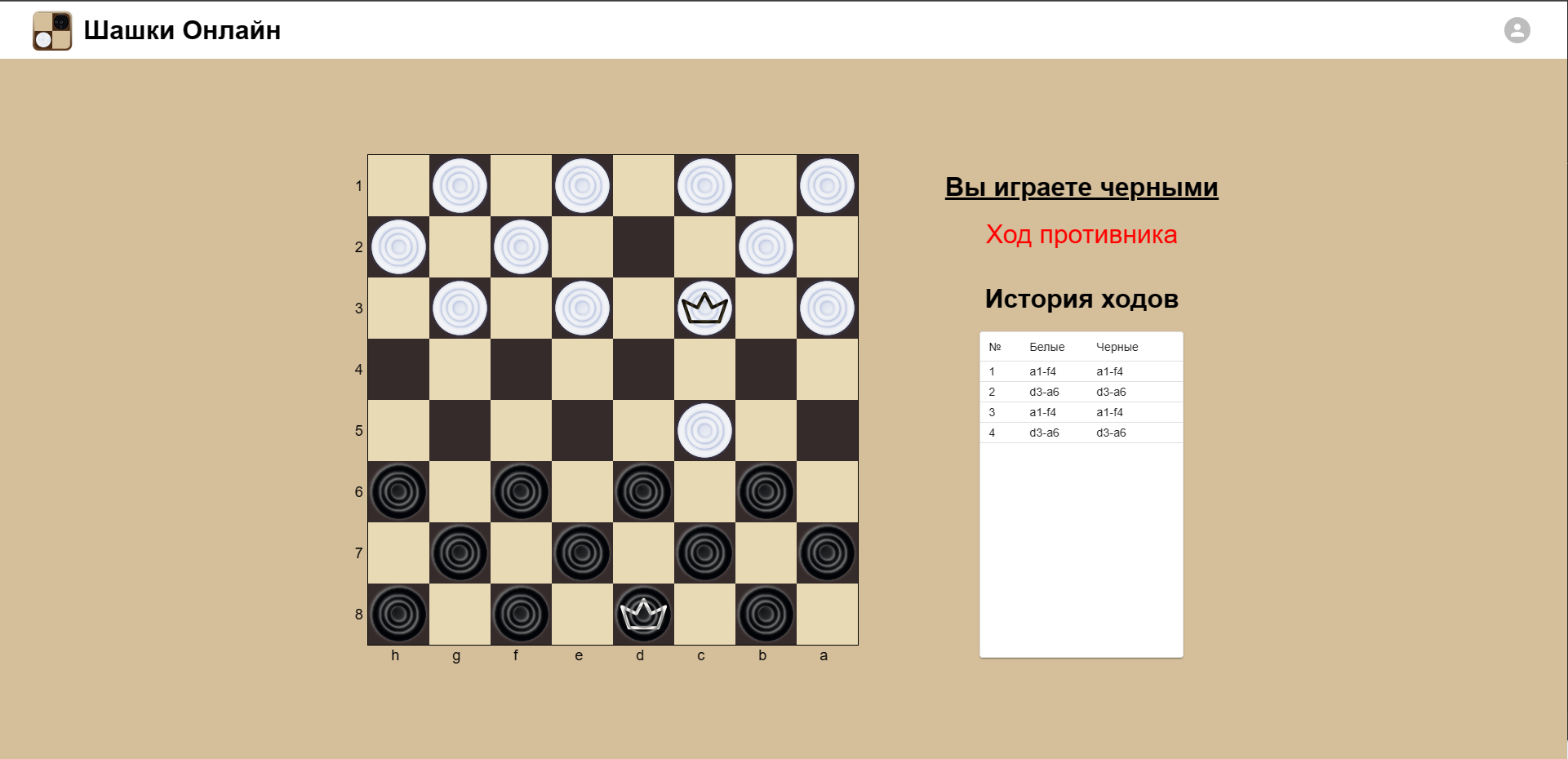


Рисунок 9 – Страница игры

Также после авторизации у пользователя появляется доступ к личному кабинету (рис. 10). В ЛК можно посмотреть кол-во сыгранных игр, побед и рейтинг, а также все завершенные игры и их результаты.



Рисунок 10 - Страница ЛК авторизованного пользователя

# Сервис WebSocket

Сервис WebSocket выполнен на языке JS с использованием библиотеки socket.io. Сервис служит для передачи уведомлений о новых открытых играх, уведомления о присоединения к игре, передачи информации о совершенном ходе игрока, создания записи хода в БД, создания и изменения записи игры в БД.

WebSocket - это протокол прикладного уровня, который обеспечивает двунаправленный обмен данными между клиентом и сервером через одно логическое соединение TCP. Для установления соединения по WebSocket одна из сторон отправляет HTTP запрос с заголовком "upgrade", и после получения ответа с заголовком "upgrade" от второй стороны, обмен информацией переходит на протокол WebSocket. Дальнейшее поддержание соединения осуществляется с помощью пакетов "ping/pong". Для закрытия соединения одна из сторон отправляет пакет с кодом 1000, и после получения подтверждения от второй стороны соединение закрывается.

На сервисе бэкенде реализован метод gRPC, который позволяет WebSocket обращаться к нему и создавать и изменять записи игр и создавать саписи ходов по протоколу protobuf. gRPC - это система удаленного вызова процедур, которая использует HTTP/2 для передачи данных и protobuf для их сериализации. Поскольку интерфейсы описываются с помощью protobuf, gRPC поддерживает множество языков программирования. В отличие от API с JSON, protobuf имеет строгую структуру и сериализуется в бинарные данные, что затрудняет их понимание человеком, но позволяет сократить объем передаваемых сообщений.

При разрыве соединения данные будут сохраняться на бэке и когда оно будет восстановлено повторно отправляться. Это обеспечивает гарантированную доставку котировок.

Работа сервиса Websocket показана на диаграмме последовательности Websocket:

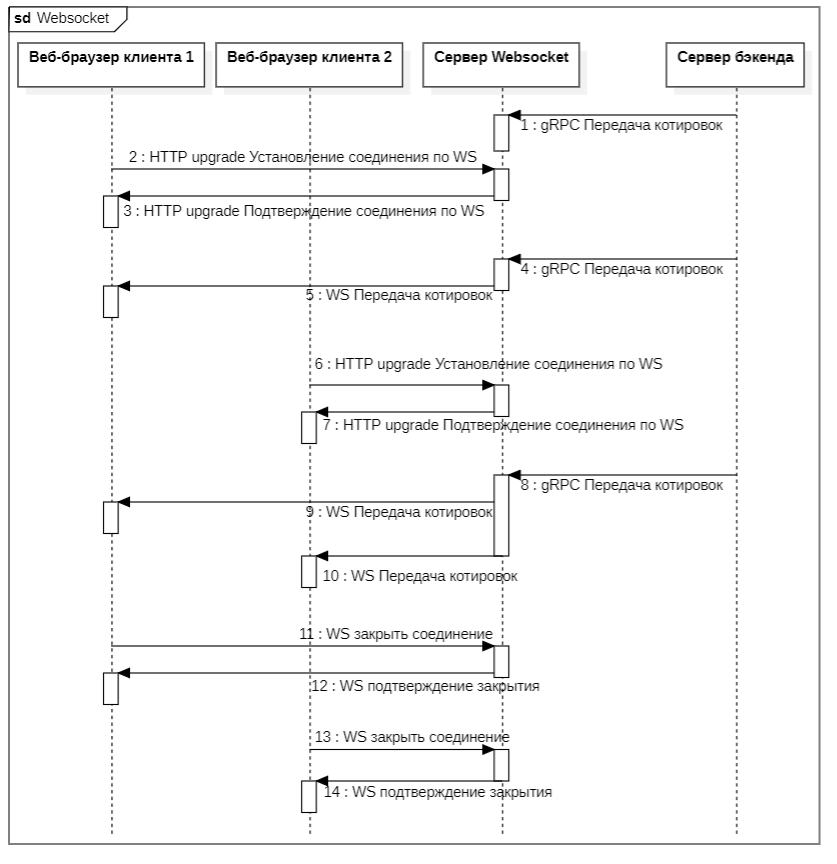


Рисунок 11 - Диаграмма последовательности Websocket

Сервис развёрнут в docker контейнере.

# Заключение

В результате выполнения курсовой работы была разработана распределённая система “шашки Онлайн”, играть по сети двум польщователям.

В процессе разработки системы были выполнены следующие задачи:

1. Техническое задание было создано.
2. Взаимодействие между компонентами системы было тщательно проработано.
3. Для облегчения коммуникации в команде разработки были разработаны диаграммы в нотации UML.
4. Был разработан пользовательский интерфейс веб-приложения.
5. Создано руководство пользователя, которое содержит информацию о взаимодействии конечного пользователя с системой.
6. Разработано руководство системного администратора, которое содержит информацию о развертывании системы.
7. Каждый компонент системы был протестирован отдельно.
8. Было проведено пробное развертывание системы.
9. Полное тестирование развернутой системы было выполнено.
10. Была проведена отладка системы.

# Список использованных источников

1. Лутц М. Изучаем Python. Т. 1 // Диалектика .- 2019.- С 832
2. Робсон Э. Изучаем программирование на JavaScript // Издательский дом ПИТЕР .- 2022 .- С 640
3. Тиленс Т. React в действии // Издательский дом ПИТЕР .- 2019 .- С 360

# Приложения