

МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)**

Факультет *Компьютерных наук*

Кафедра *программирования и информационных технологий*

Сервис настольных онлайн-игр «Настольный сервер»

Курсовой проект

09.03.04 Программная инженерия

Преподаватель _____ В.С. Тарасов __.__.20__

Обучающийся _____ А.А. Антонов, 3 курс, д/о

Обучающийся _____ А.В. Тарасов, 3 курс, д/о

Обучающийся _____ А.А. Кусакин, 3 курс, д/о

Воронеж 2021

Содержание

| | |
|---|----|
| Содержание | 2 |
| Введение | 4 |
| 1. Постановка задачи | 5 |
| 1.1 Цель создания системы | 5 |
| 1.2 Сфера использования системы | 5 |
| 1.3 Требования к создаваемой системе | 5 |
| 1.4 Задачи, решаемые в процессе разработки | 6 |
| 2. Анализ предметной области | 7 |
| 2.1 Целевая аудитория | 7 |
| 2.2 Описание предметной области | 9 |
| 2.3 Пользователи системы | 10 |
| 2.4 Границы функциональности | 11 |
| 2.5 Анализ существующих решений | 11 |
| 2.6 Воронки конверсии | 17 |
| 2.6.1. Цель «Регистрация» | 19 |
| 2.6.2. Цель «Создание игровой комнаты» | 19 |
| 2.6.3. Цель «Присоединение к игровой комнате по ссылке» | 20 |
| 2.7 Анализ задачи | 21 |
| 2.7.1. Варианты использования приложения | 21 |
| 2.7.2. Взаимодействие компонентов системы | 21 |
| 2.7.3. Варианты состояния системы | 29 |
| 2.7.4. Варианты действий в системе | 31 |
| 2.7.5. Развертывание приложения | 31 |
| 2.7.6. Диаграмма классов | 32 |
| 2.7.7. Диаграмма объектов | 33 |
| 2.7.8. IDEF0 | 34 |
| 3. Архитектура приложения | 36 |
| 3.1 Выбор средств реализации | 36 |
| 3.2 Архитектура системы | 37 |
| 3.3 Структура базы данных | 39 |

| | | |
|------|--|----|
| 3.4 | Конфигурация приложения | 41 |
| 3.5 | Работа с веб-сокетами | 42 |
| 3.6 | Взаимодействие между серверной и клиентской частями системы. Swagger..... | 43 |
| 4. | Интерфейс приложения | 44 |
| 4.1 | Основное и дополнительное меню | 44 |
| 4.2 | Страница регистрации..... | 45 |
| 4.3 | Страница авторизации | 45 |
| 4.4 | Главная страница приложения | 46 |
| 4.5 | Страница игры | 47 |
| 4.6 | Окно настроек игровой комнаты | 48 |
| 4.7 | Страница игровой комнаты | 48 |
| 4.8 | Страница с игровым полем..... | 49 |
| 4.9 | Страница личного кабинета | 50 |
| 4.10 | Страница идущих игр | 51 |
| 4.11 | Страница истории..... | 51 |
| 5. | Тестирование..... | 53 |
| 5.1 | Дымовое тестирование..... | 53 |
| 5.2 | UI тестирование | 54 |
| 5.3 | Usability тестирование..... | 59 |
| | Заключение | 61 |
| | Список использованных источников | 62 |

Введение

Настольные игры с давних времен занимали значимое место в жизни людей. Развивающие воображение, память, образ мышления и внимательность, они становятся объектом внимания людей всех возрастов, предлагая для размышления различные задачи и выборы, которые порой вовсе не выглядят простыми. В наше время, когда большинство людей нуждается в отдыхе, отвлекающем от рутины привычных дней, роль таких игр заметно выросла. Уникальные миры и сюжеты, захватывающие игроков, интересные решения и неожиданные повороты с каждым годом привлекают в сообщество игроков все новых и новых людей.

В свою очередь, возможность играть в любимые игры с друзьями и товарищами, которые находятся далеко, становится мечтой почти каждого настольщика. Во времена, когда отсутствие компьютера уже редкость, а сотни компаний предлагают десятки тысяч игр в самых различных жанрах, в которые можно играть как в одиночку, так и в компании друзей, вопрос, почему до сих пор никто не заинтересовался разработкой виртуальных настольных игр, ставит в небольшой тупик множество людей. Безусловно, в настоящий момент активно развиваются виртуальные столы для шахмат и ММО-игр. Так, партии по культовым играм вроде D&D проводят, собирая группы людей со всего мира.

Однако менее популярные игры, к сожалению, лишены такой поддержки. В лучшем случае, виртуальную версию желаемой игры удастся найти на каком-нибудь левом сайте, заваленном рекламой, но ни один из них не представляет собой простой инструмент, позволяющий не разбираясь в требуемых регистрациях и недостающем программном обеспечении отдохнуть в компании друзей за любимой игрой.

Данный курсовой проект направлен на разработку и создание платформы, поддерживающей работу пошаговых настольных онлайн-игр и позволяющей пользователю окунуться в игровой мир в компании своих товарищей.

1. Постановка задачи

Необходимо разработать веб-приложение, реализующее функциональность игровой платформы. Система должна специализироваться на жанре настольных игр. Система не должна зависеть от сторонних платформ и быть легко масштабируемой.

1.1 Цель создания системы

Система разрабатывается в следующих целях:

- создание сервиса для игры в настольные игры в режиме онлайн;
- осуществление взаимодействия пользователей, в том числе голосовое общение;
- создание сервиса, независимого от сторонних платформ и плагинов (например, Flash Player);
- создание сервиса, имеющего низкий порог вхождения для новых пользователей.

1.2 Сфера использования системы

Повседневная жизнь.

1.3 Требования к создаваемой системе

Разрабатываемое веб-приложение должно предоставлять следующие возможности:

1. регистрация и авторизация пользователей;
2. редактирование профиля пользователя;
3. обеспечение возможности онлайн-игры в настольные игры группе людей;
4. приглашение пользователем неавторизованных гостей для принятия участия в игре;
5. обеспечение возможности общения игроков между собой с помощью голосовой связи в процессе игры;

6. возможность пользователя просматривать правила игр и статистику по сыгранным партиям;
7. сбор статистики игр пользователей;
8. обеспечение возможности дальнейшего расширения библиотеки игр.

1.4 Задачи, решаемые в процессе разработки

В процессе разработки системы необходимо решить следующие задачи:

1. провести анализ рынка с целью выявления достоинств и недостатков схожих по функционалу систем;
2. спроектировать систему с учетом информации, полученной ранее в ходе анализа;
3. разработать базу данных;
4. разработать Back-end часть приложения;
5. разработать Front-end часть приложения;
6. создать связь между Front-end и Back-end частями приложения;
7. создать и развернуть первую тестовую игру («Каркассон»);
8. описать процесс разработки и результат.

2. Анализ предметной области

2.1 Целевая аудитория

Сервис настольных игр подойдет для всех, кто любит игры такого формата. Согласно проведенному опросу, в котором участвовало 40 респондентов, около 70% людей любят проводить свободное время, общаясь с друзьями или играя в какие-либо онлайн-игры (Рис.1.).

Чем Вы предпочитаете заниматься в свободное время в интернете?

40 ответов

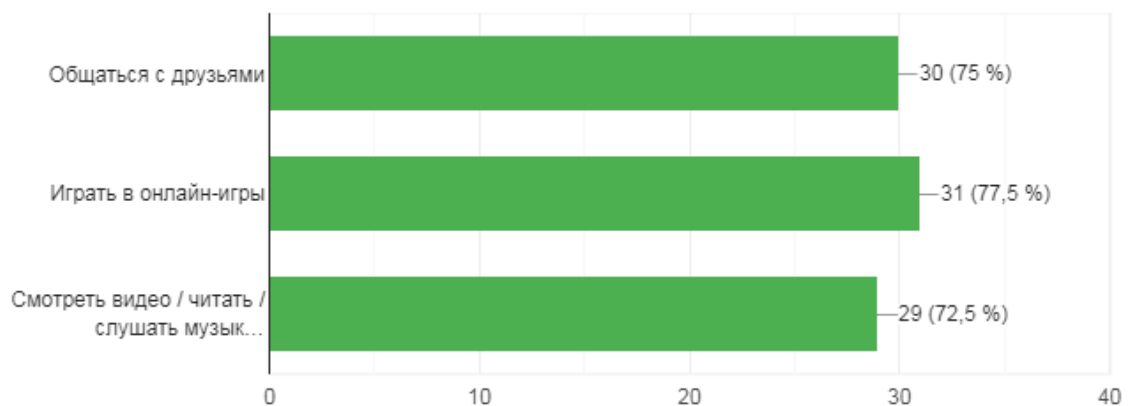


Рис. 1. Результаты опроса

При этом только 15% респондентов не интересуют игры настольного формата вообще, 22,5% активно используют существующие веб-приложения, 23% предпочитают играть с друзьями при личной встрече, и 37,5% не играют, так как у них нет возможности сыграть с друзьями (Рис.2.).

Играете ли Вы в игры настольного формата (карты, шахматы, монополия и т.д.)?

40 ответов



Рис. 2. Результаты опроса

Также из 27 респондентов, указавших, что они играют в онлайн-игры (любого формата), 23 человека отметили, что вместе с игрой они используют сторонние сервисы, предоставляющие возможность голосовой связи (Skype, Discord или другие). 4 человека не пользуются голосовой связью. И ни один респондент не отметил, что он пользуется голосовой связью, которую предоставляет сервис самой онлайн-игры (Рис.3.).

Если Вы играете в онлайн-игры с друзьями, выберите подходящий Вам вариант:

27 ответов



Рис. 3. Результаты опроса

Данные опроса показывают, что, хоть сервисы настольных онлайн-игр достаточно распространены, они не предоставляют достаточного функционала для игры с друзьями: во-первых, не всегда можно пригласить

друга на игру (ему необходимо будет зарегистрироваться; некоторые сервисы предоставляют эту возможность платно), во-вторых, для голосовой связи с друзьями пользователю необходимо запускать сторонние приложения, а связаться с незнакомыми игроками вообще достаточно сложно и явно будет отвлекать от самой игры (подробнее об аналогичных приложениях в пункте 2.5).

Таким образом, разрабатываемое веб-приложение ориентировано на большинство пользователей, играющих в настольные игры и имеющих компьютер с доступом в сеть Интернет.

2.2 Описание предметной области

Разрабатываемая система является платформой для размещения онлайн-игр, использующих технологии HTML, CSS и JavaScript. С развитием информационных технологий и сети Интернет такие платформы получили большое распространение и популярность, так как они предлагают большое количество различных игр, причем пользователю не нужно скачивать и устанавливать понравившуюся игру себе на компьютер.

Однако большинство таких сервисов использовали и продолжают использовать технологию Flash, которая была признана небезопасной. Кроме того, современные технологии, такие как HTML5, позволяют отказаться от использования устаревших программных средств. Поэтому важным требованием к разрабатываемой системе является независимость от сторонних плагинов и платформ.

Разрабатываемая платформа ориентирована на игры настольного формата. Сервисы, предоставляющие возможность сыграть в такие игры, встречаются относительно редко.

Кроме того, разрабатываемая система должна предоставить пользователям дополнительные возможности: голосовое общение в процессе игры и возможность приглашения незарегистрированных участников с помощью ссылки. Это существенно отличает разрабатываемый сервис от

аналогов, так как такие возможности из них предоставляют единицы, что должно повысить комфорт использования, а значит и популярность проекта.

2.3 Пользователи системы

Система предназначена для участия пользователя в онлайн-играх и общения с другими игроками.

Для взаимодействия с разрабатываемой системой выделяют следующие типы пользователей: гость и пользователь.

Гость (неавторизованный в системе пользователь) обладает следующими параметрами:

- ID неавторизованного пользователя для текущей партии.

В процессе взаимодействия с системой гость имеет следующие возможности:

- принять участие в игре по приглашению пользователя;
- наблюдать за ходом игры по приглашению участвующего в ней пользователя;
- переговариваться с другими участниками партии с помощью голосовой связи;
- просматривать правила игры;
- просматривать результаты партии, в которую гость был приглашен в качестве участника или наблюдателя.

Пользователь (авторизованный в системе пользователь) обладает следующими параметрами:

- имя пользователя;
- электронная почта;
- пароль.

В процессе взаимодействия с системой пользователь имеет следующие возможности:

- просмотр правил игры;
- изменение персональных данных в личном кабинете;
- создание новой игры с выбранными параметрами;
- приглашение гостя в качестве участника игры или наблюдателя;
- общение с другими участниками партии с помощью голосовой связи;
- просмотр результатов партии;
- просмотр общей статистики.

2.4 Границы функциональности

Данная система реализует строго определенные возможности пользователя:

- создание личного кабинета;
- изменение личных данных;
- просмотр существующих игр;
- создание новой игры с выбранными параметрами;
- приглашение гостя для участия в игре;
- общение с другими участниками игры с помощью голосовой связи;
- просмотр правил игры, результатов сыгранной партии, статистику по всем сыгранным партиям.

Также система реализует сбор статистики действий пользователя для дальнейшего анализа и поддерживает расширение библиотеки игр путем добавления новых без внесения изменений в существующие модули системы.

2.5 Анализ существующих решений

Большинство сервисов, предлагающие возможность поиграть в какие-либо настольные игры, устарели и требуют наличия Flash Player, который на данный момент использовать не рекомендуется.

Чтобы сделать анализ аналогов более конкретным, рассмотрим реализацию игры «Каркассон», как первого продукта на нашей платформе.

Большинство сервисов, найденных в ходе анализа, имеют большое количество рекламы других игр и сайтов с совершенно другой тематикой. Часто реклама становится слишком навязчивой и появляется над активными элементами страницы, существенно затрудняя процесс игры. Интерфейс таких сервисов бывает рассеянным и хаотичным. Кроме того, некоторые из этих сайтов при запуске игры встречают пользователя низкокачественной, но очень громкой музыкой, которую далеко не всегда можно выключить. Только очень настойчивый и терпеливый пользователь сможет приступить к игре.

Рассмотрим некоторые найденные аналоги подробнее.

1) Сервис «ИГРОУТКА»

<https://igrouotka.net/igry-strategii/24127-karkasson.html>

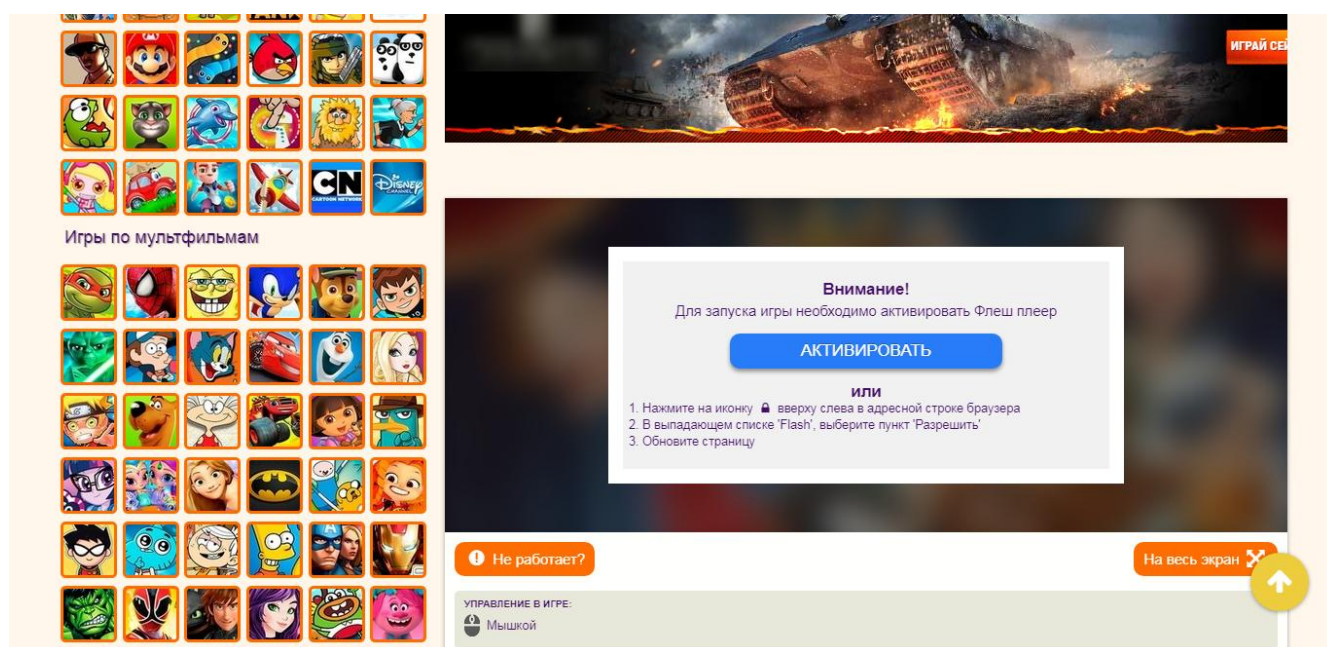


Рис. 4. Интерфейс сервиса «ИГРОУТКА»

Достоинства:

- возможность играть без регистрации;

Недостатки:

- игра работает на технологии Flash Player;
- на странице присутствуют 9 рекламных баннеров сторонних сайтов, некоторые из которых размером превосходят окно самой игры;
- на странице присутствует огромное количество иконок других игр, разбросанные по всей странице, отвлекающих от игры;
- отвлекающий внимание от основного контента интерфейс, неправильное расположение элементов страницы (окно игры находится в нижней части страницы, после блока ссылок на другие игры и рекламы);
- играть можно только с компьютером, пригласить других игроков невозможно.

2) Сервис «More-Games.Ru»

<https://more-games.ru/online/5221>

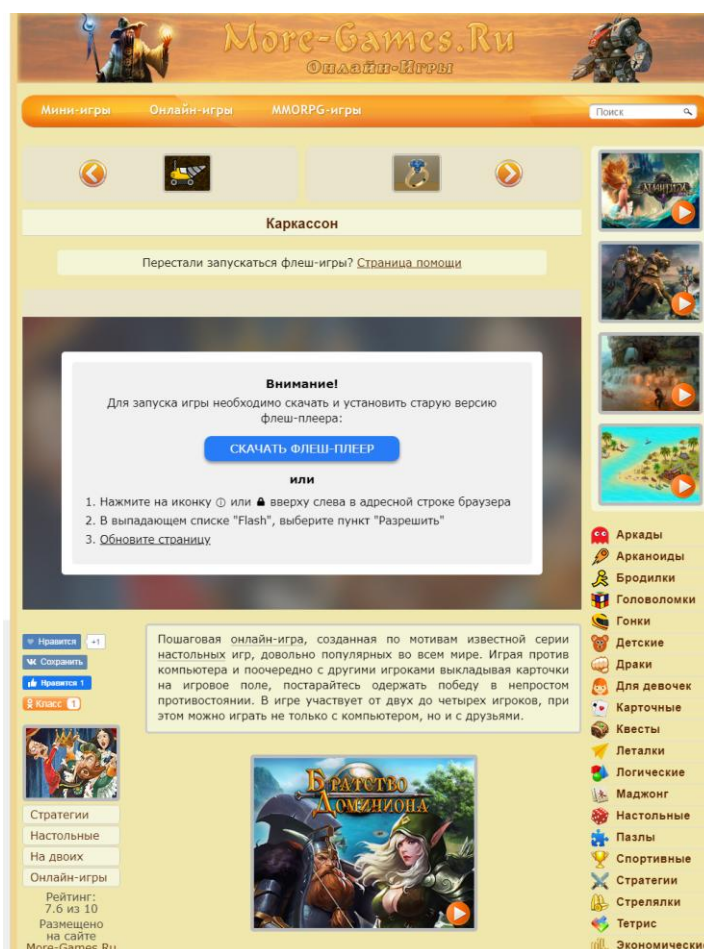


Рис. 5. Интерфейс сервиса «More-Games.Ru»

Достоинства:

- возможность играть без регистрации;
- отсутствует навязчивая и слишком отвлекающая внимание реклама;
- логичное и удобное для пользователя расположение элементов страницы;
- небольшое количество ссылок на другие игры;
- не отвлекающий внимание дизайн.

Недостатки:

- игра работает на технологии Flash Player;
- играть можно только с компьютером, пригласить других игроков невозможно.

3) Сервис «BoardGameArena»

<https://boardgamearena.com/gamepanel?game=carcassonne>

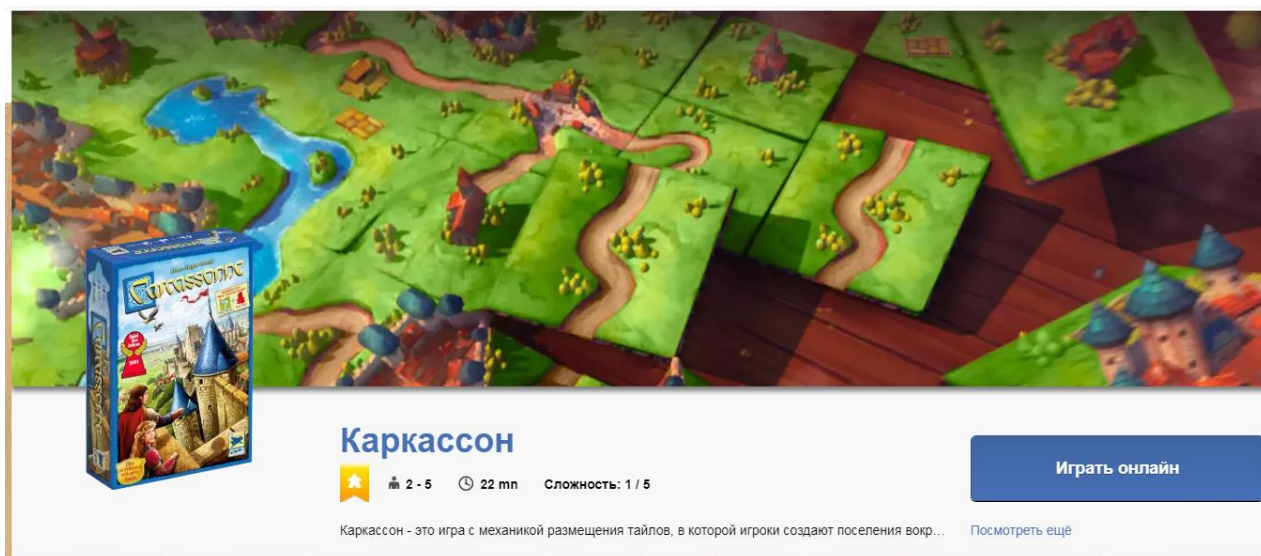


Рис. 6. Интерфейс сервиса «BoardGameArena»

Достоинства:

- полное отсутствие рекламы;
- приятный и удобный интерфейс пользователя;
- возможность игры с другими пользователями сервиса;
- возможность пригласить друзей на игру;
- возможность связи через видео- и аудио-чат.

Недостатки:

- для того чтобы принимать участие в игре, необходимо стать членом клуба (платно);
- сообщение о том, что возможность игры предоставляется платно, появляется после регистрации и настройки профиля;
- в игре могут участвовать только члены клуба.

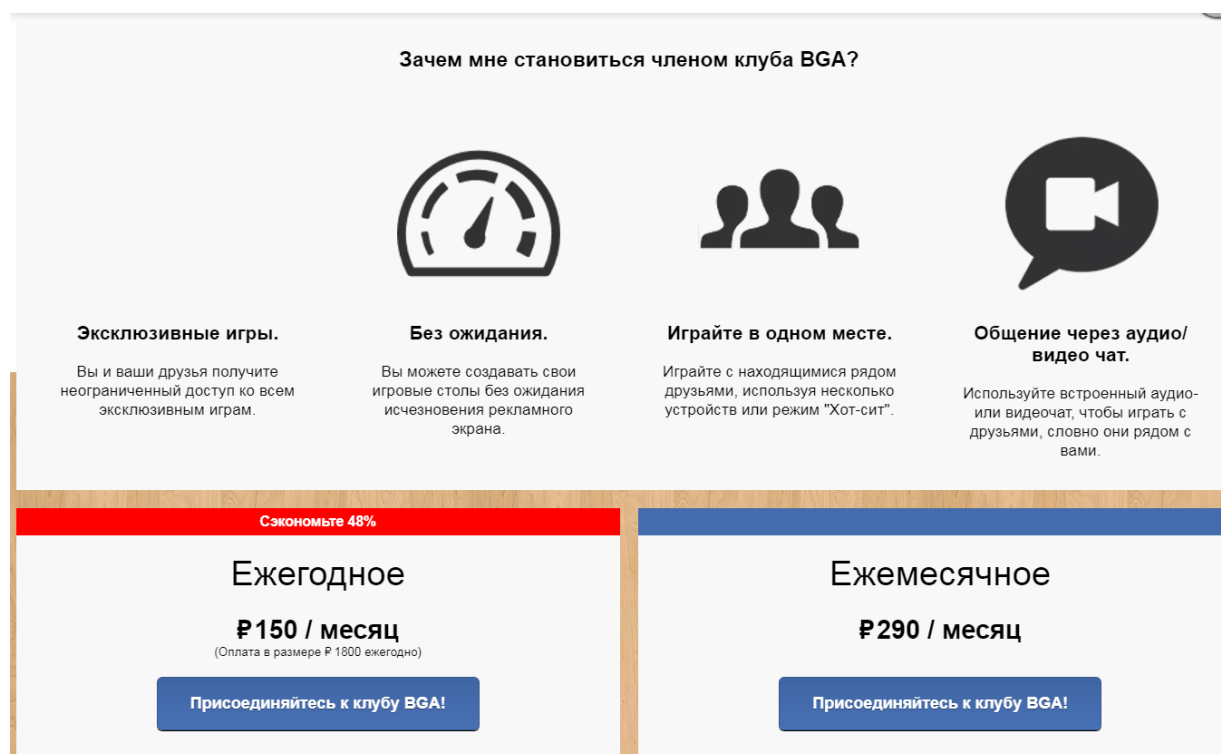


Рис. 7. Предложение покупки в приложении «BoardGameArena»

Таким образом, можно сказать, что среди аналогов, представленных в сети Интернет, на данный момент нет ни одного сервиса, способного предоставить пользователю быструю возможность сыграть в любимую игру, поделиться ее результатами с товарищами или провести вечер в компании друзей, общаясь с ними внутри игры.

В результате анализа существующих решений был сформирован перечень основных недостатков аналогичных веб-приложений, а также продуманы варианты их устранения:

Таблица 1. – Недостатки аналогов и варианты их устранения

| Недостаток | Решение |
|--|-----------------------------------|
| Игра использует технологию Flash Player, использование которой на данный момент не рекомендуется | Использование HTML5 и Angular CLI |

Продолжение таблицы 1.

| Недостаток | Решение |
|--|--|
| Слишком навязчивая реклама сторонних сайтов | Неиспользование рекламных баннеров, которые перекрывают активные элементы страницы или слишком отвлекают пользователя от основного контента |
| Хаотичный и неудобный интерфейс | Интерфейс должен позволять легко воспринимать основную информацию сайта. Также не стоит добавлять на одну страницу большое количество ссылок, иконок других игр или другую ненужную информацию |
| Слишком громкое или низкокачественное звуковое сопровождение, невозможность его отключения | Возможность настройки громкости звука в процессе использования приложения |

2.6 Воронки конверсии

Яндекс.Метрика собирает информацию о взаимодействии пользователя с сайтом и фиксирует достижение целей. Затем, на основе собранных данных рассчитываются целевые метрики. Цель - это действие пользователя, в котором заинтересован владелец сайта.

Созданные цели позволяют отслеживать события на сайте (нажатие кнопки, заполнение формы и пр.), при выполнении которых не меняется URL страницы. Информация о достижении такой цели передается в Яндекс.Метрику с помощью JavaScript, что позволяет отслеживать практически любые произвольные события.

Достижение цели отслеживается при выполнении условия, заданного в параметрах конкретной цели.

https://metrika.yandex.ru/stat/conversion_rate?group=day&period=week&accuracy=1&id=80679118

Метрики с конверсиями для различных сценариев представлены на рисунке 8.

Общая сводка по просмотрам и посещениям страниц представлена на рисунке 9.

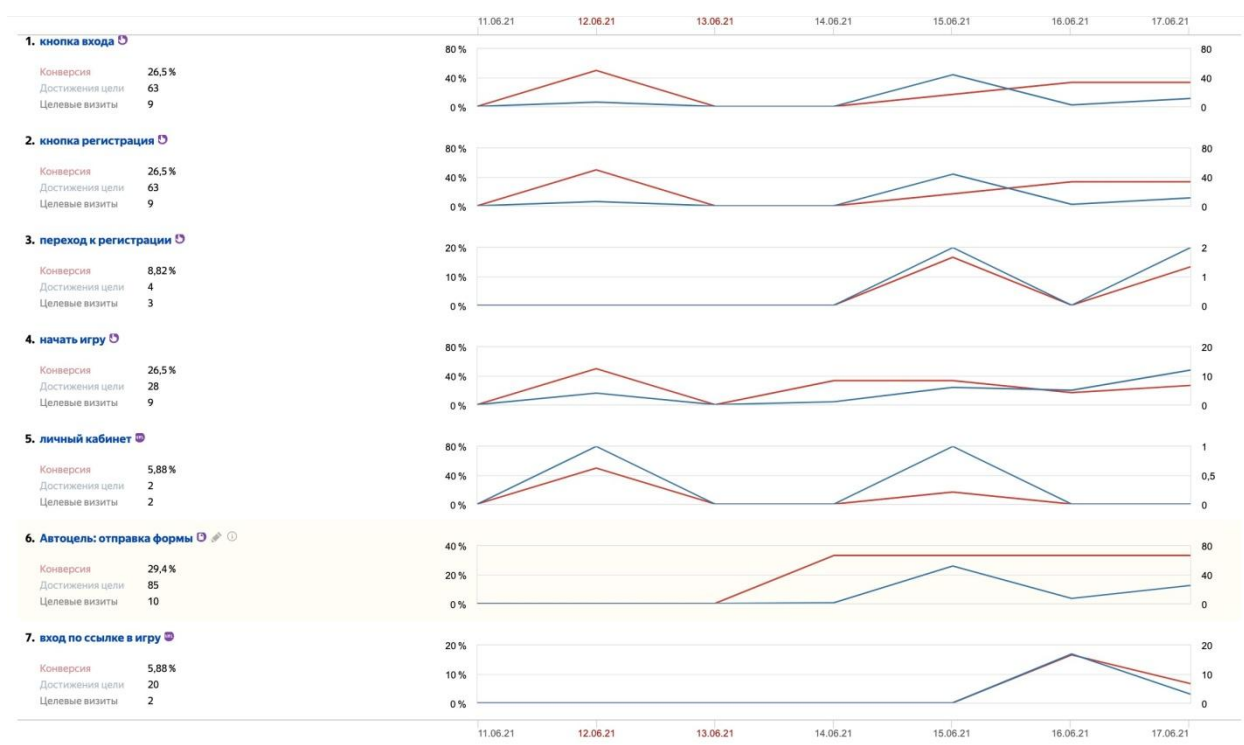


Рис. 8. Конверсии Яндекс.Метрики для основных сценариев

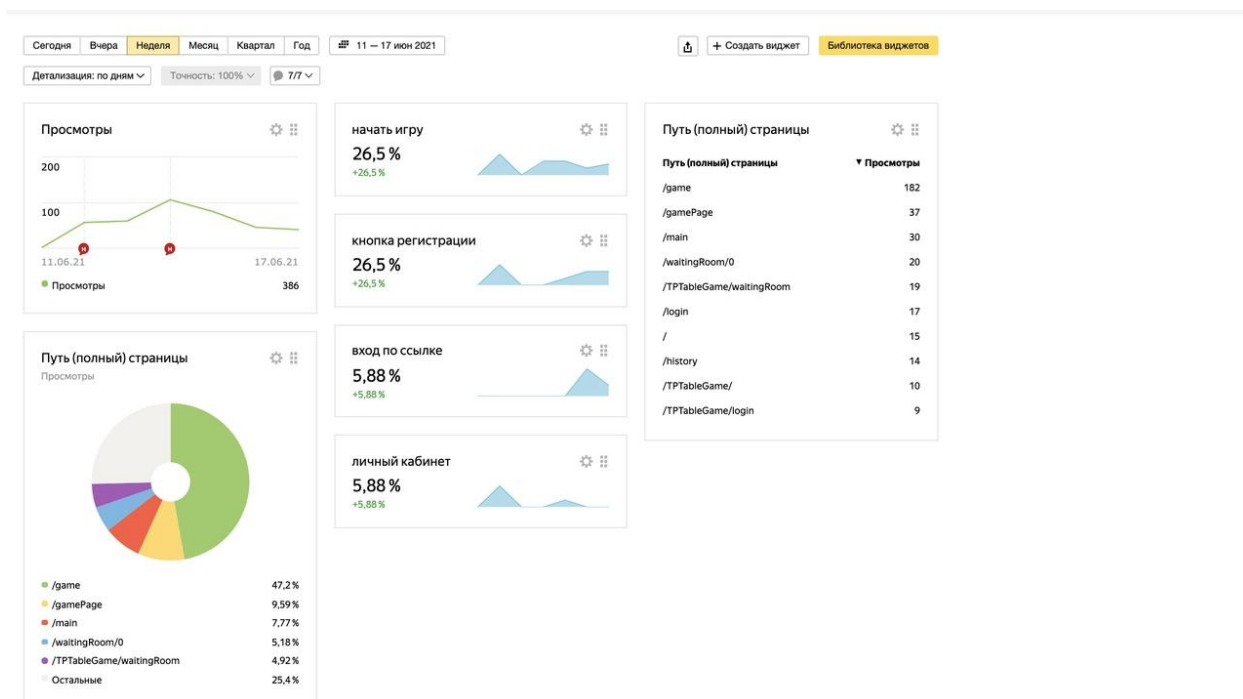


Рис. 9. Общая сводка

2.6.1. Цель «Регистрация»

Использование Яндекс.Метрики для учета количества регистраций позволит отследить прирост новых зарегистрированных пользователей. Это также важно учитывать, поскольку сервис предоставляет возможности игры по приглашению без регистрации.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- открыть страницу регистрации;
- ввести данные в регистрационную форму;
- нажать кнопку «Войти».

Данные по конверсии и достижению целей для данного сценария представлены на рисунке 8, сценарий 3.

2.6.2. Цель «Создание игровой комнаты»

Учет создания новых игровых комнат позволит определить популярность игры в общем или за какой-либо промежуток времени, а так же

количество пользователей-организаторов. Также можно учитывать количество пользователей, которые не смогли до конца выполнить условия данной цели.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- нажать «Играть» на выбранной игре в библиотеке;
- ознакомившись с правилами, нажать «Играть»;
- ввести данные в форму настройки игровой комнаты;
- создать игровую комнату, нажав «Начать».

Данные по конверсии и достижению целей для данного сценария представлены на рисунке 8, сценарий 4.

2.6.3. Цель «Присоединение к игровой комнате по ссылке»

Учет присоединившихся по ссылке пользователей позволит оценить популярность конкретной игры, количество пользователей-организаторов, приглашающих участников с помощью ссылки, а также количество незарегистрированных участников, пользующихся сервисом (именно играющих в игры, а не просто просматривающих страницы), т.е. популярность сервиса в целом.

Для выполнения условий данной воронки пользователю необходимо совершить следующие действия:

- перейти по ссылке приглашения;
- присоединиться к игре.

Данные по конверсии и достижению целей для данного сценария представлены на рисунке 8, сценарий 7.

2.7 Анализ задачи

2.7.1. Варианты использования приложения

Система реализует строго определенные возможности. По взаимодействию с системой выделяются два типа пользователей: пользователь и гость (более подробно ознакомиться с типами пользователей можно в пункте 2.3 «Пользователи системы»), возможности которых проиллюстрированы на рисунке 8.



Рис. 10. Диаграмма прецедентов (Use-case диаграмма)

2.7.2. Взаимодействие компонентов системы

На рисунке 9 показана диаграмма последовательности для взаимодействия пользователя с системой для следующих прецедентов:

- авторизация;
- просмотр библиотеки игр;
- просмотр идущих игр;

- выбор игры, просмотр правил и результатов предыдущих партий пользователя, создание новой игровой комнаты (учитывается Яндекс.Метрикой);
- просмотр результатов всех сыгранных партий (истории);
- просмотр личного кабинета, изменение личных данных;
- присоединение к игровой комнате и к игре.

Диаграммы взаимодействий для этих действий представлены на рисунках 11–17.

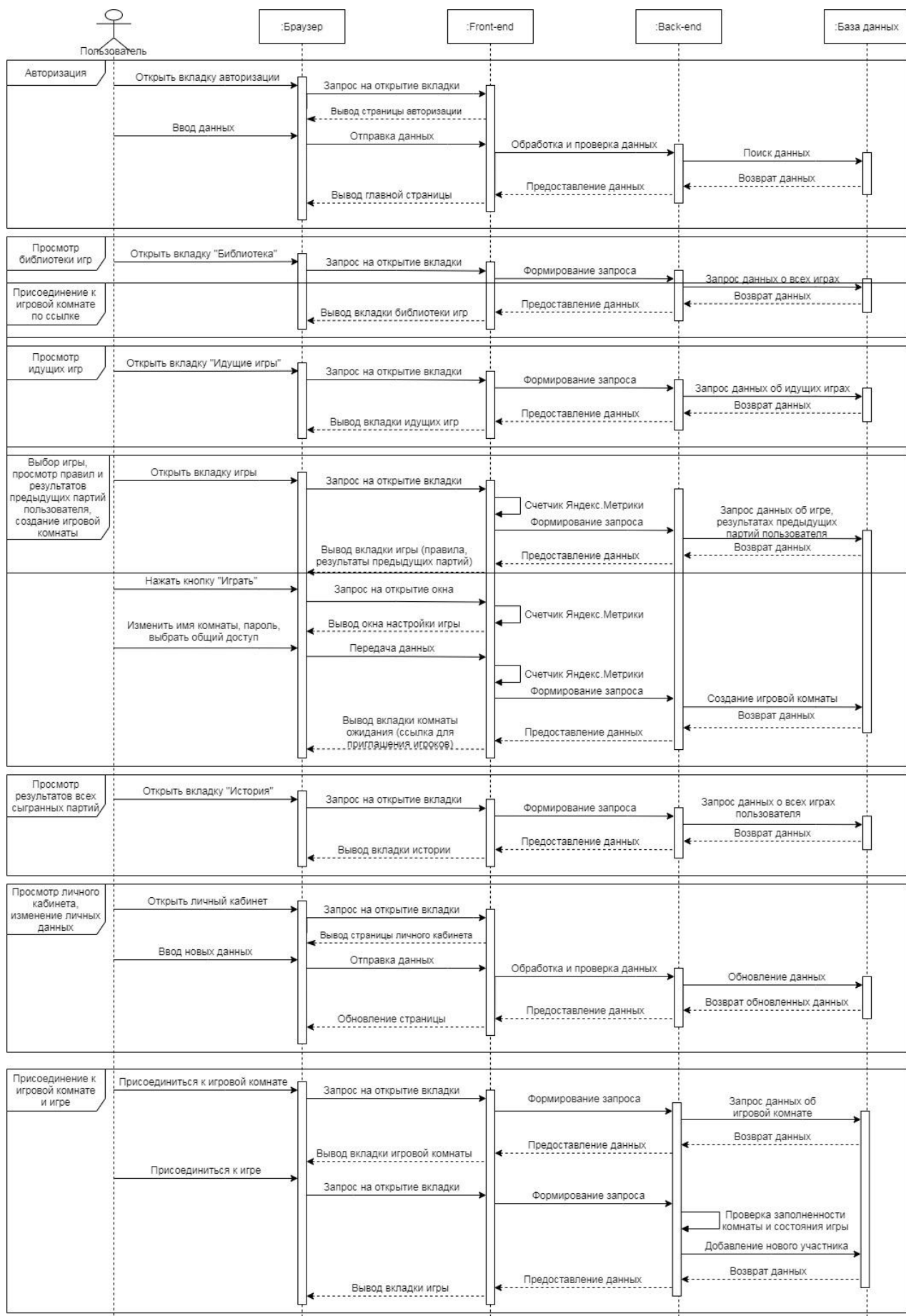


Рис. 11. Диаграмма последовательностей для пользователя



Рис. 12. Диаграмма взаимодействий для прецедента «Авторизация»

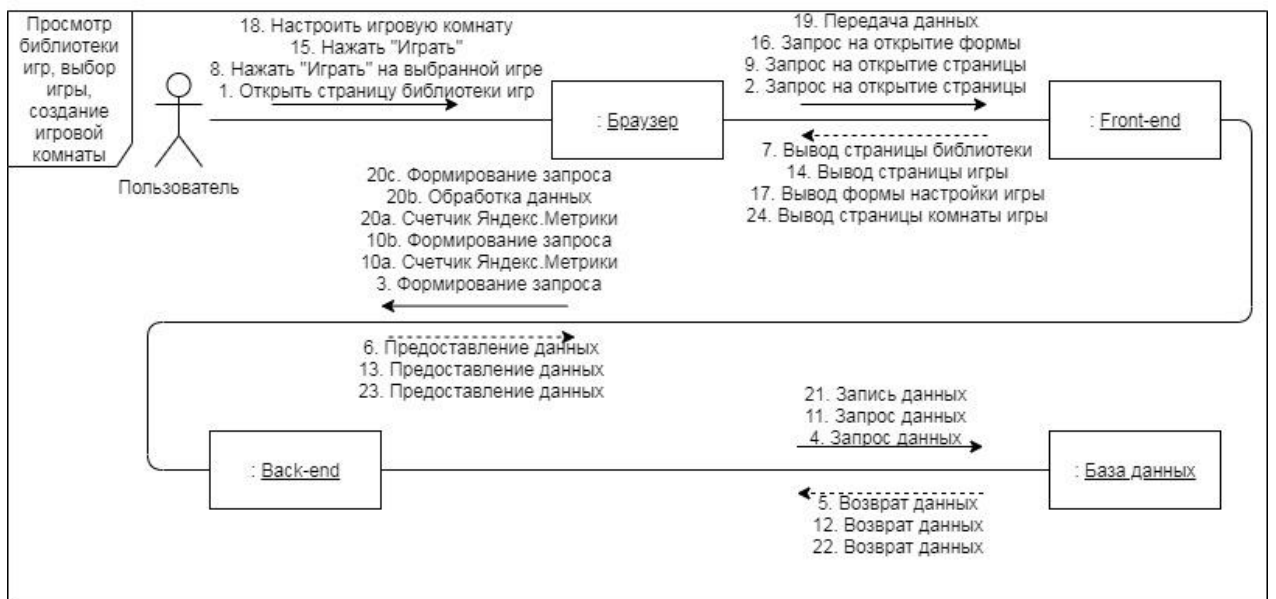


Рис. 13. Диаграмма взаимодействий для прецедента «Просмотр библиотеки» и «Начало игры»



Рис. 14. Диаграмма взаимодействий для прецедента «Просмотр идущих игр»



Рис. 15. Диаграмма взаимодействий для прецедента «Просмотр личного кабинета, изменение личных данных»



Рис. 16. Диаграмма взаимодействий для прецедента «Просмотр результатов всех сыгранных партий»

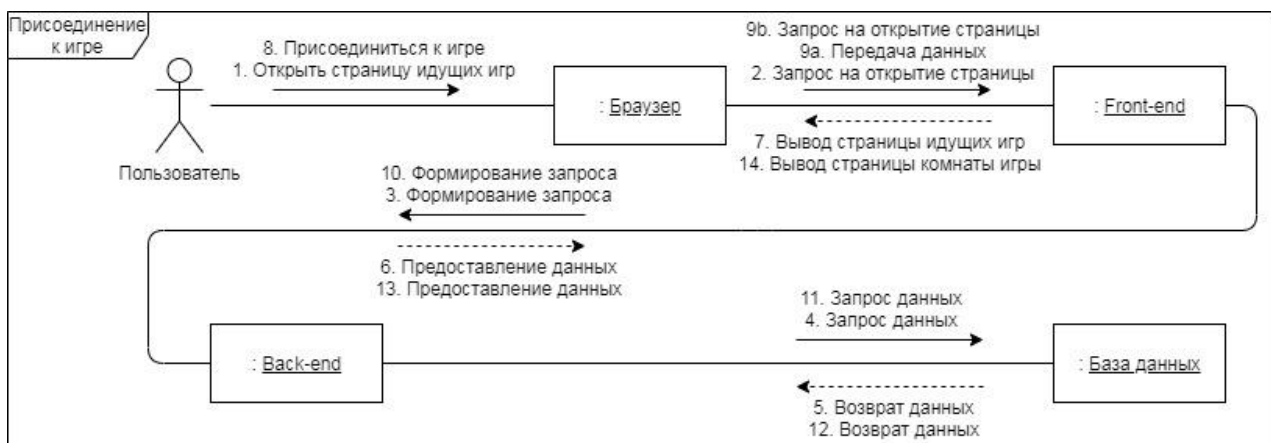


Рис. 17. Диаграмма взаимодействий для прецедента «Присоединение к игровой комнате и к игре»

На рисунке 16 показана диаграмма последовательности для взаимодействия гостя с системой для следующих прецедентов:

- регистрация (учитывается Яндекс.Метрикой);
- присоединение к игровой комнате по ссылке (учитывается Яндекс.Метрикой);

Диаграммы взаимодействий для этих действий представлены на рисунках 18–20 соответственно.

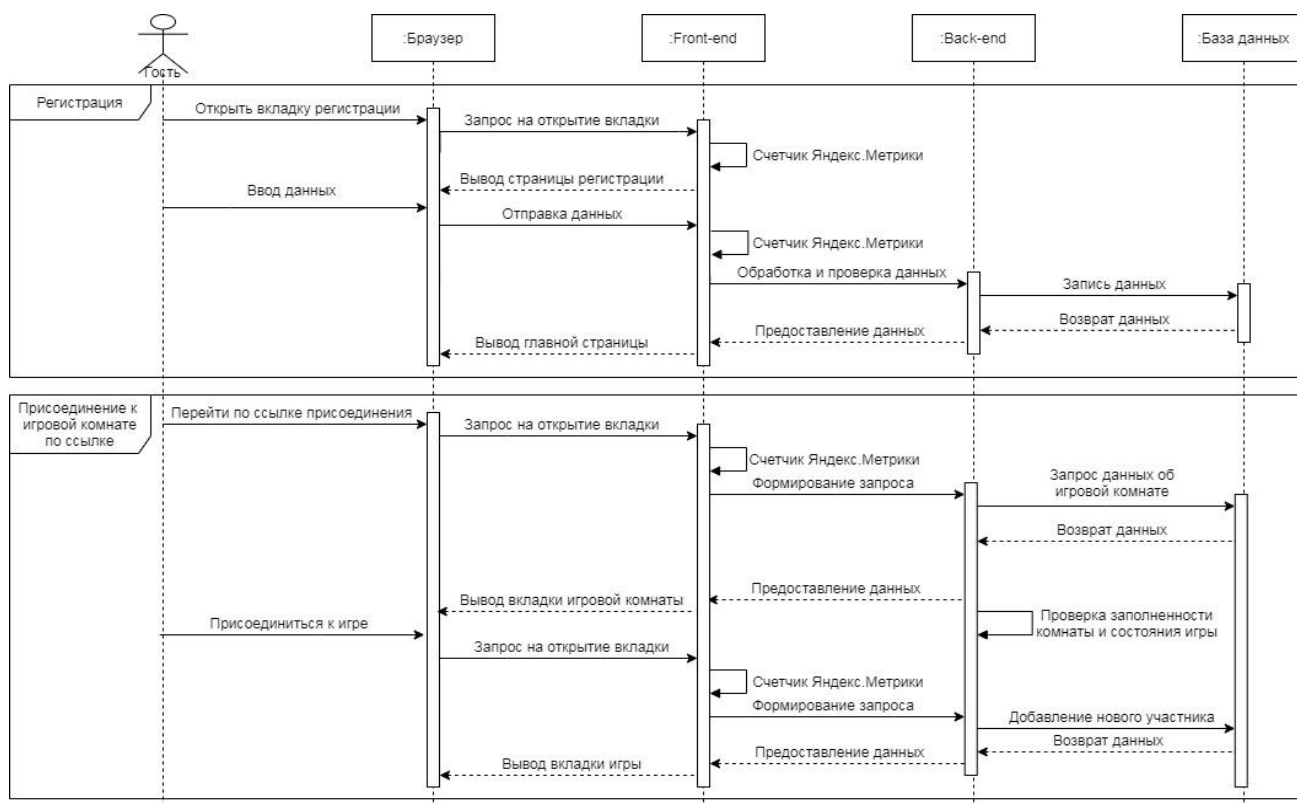


Рис. 18. Диаграмма последовательностей для гостя



Рис. 19. Диаграмма взаимодействий для прецедента «Регистрация»

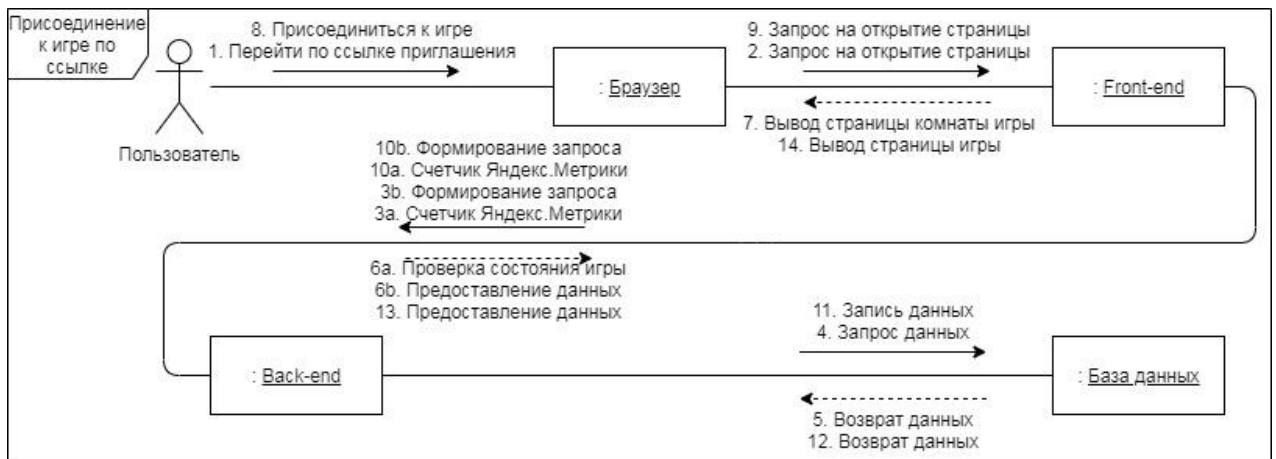


Рис. 20. Диаграмма взаимодействий для прецедента «Присоединение к игре по ссылке»

На рисунках 21-23 показаны диаграммы последовательностей для взаимодействий с системой, проходящих одинаково как для пользователя, так и для гостя:

- игра, завершение игры, просмотр результатов;
- присоединение к голосовому чату, отключение от голосового чата.

Диаграммы взаимодействий для этих действий представлены на рисунках – соответственно.

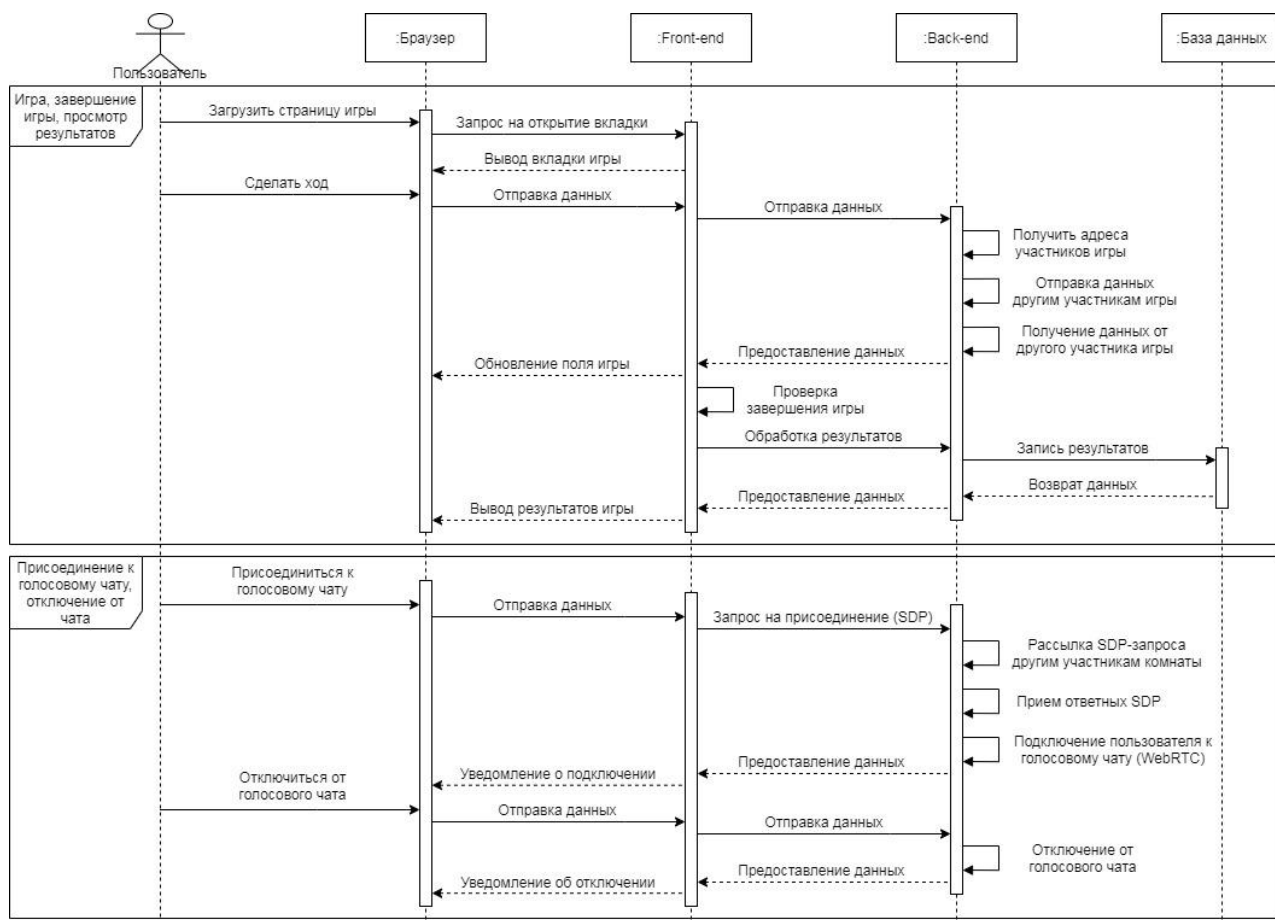


Рис. 21. Диаграмма последовательностей для взаимодействий, проходящих одинаково для пользователя и гостя

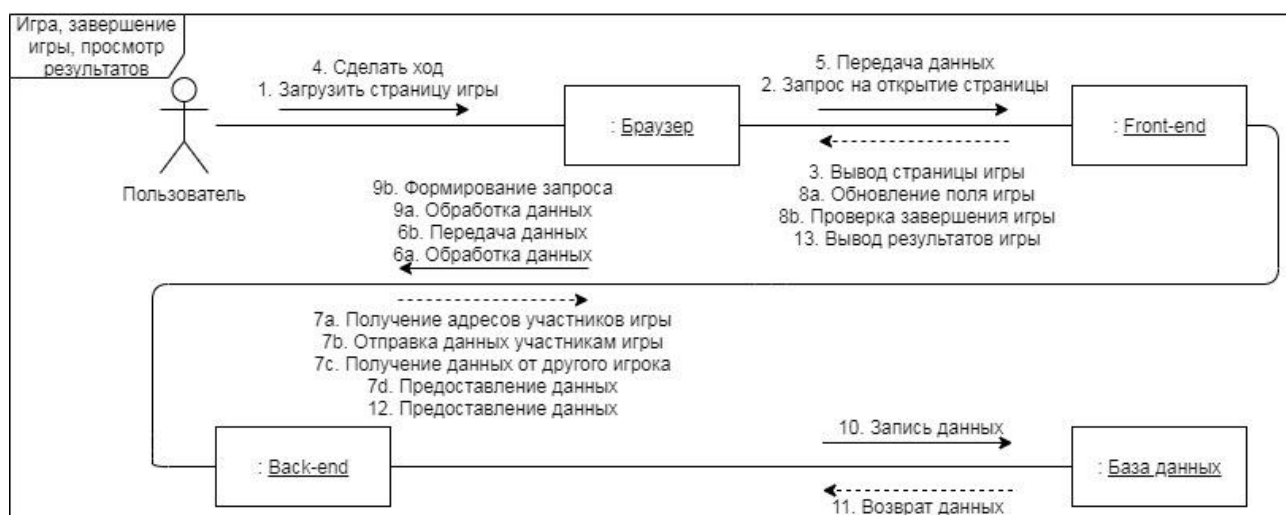


Рис. 22. Диаграмма взаимодействий для прецедентов «Игра», «Завершение игры» и «Просмотр результатов»

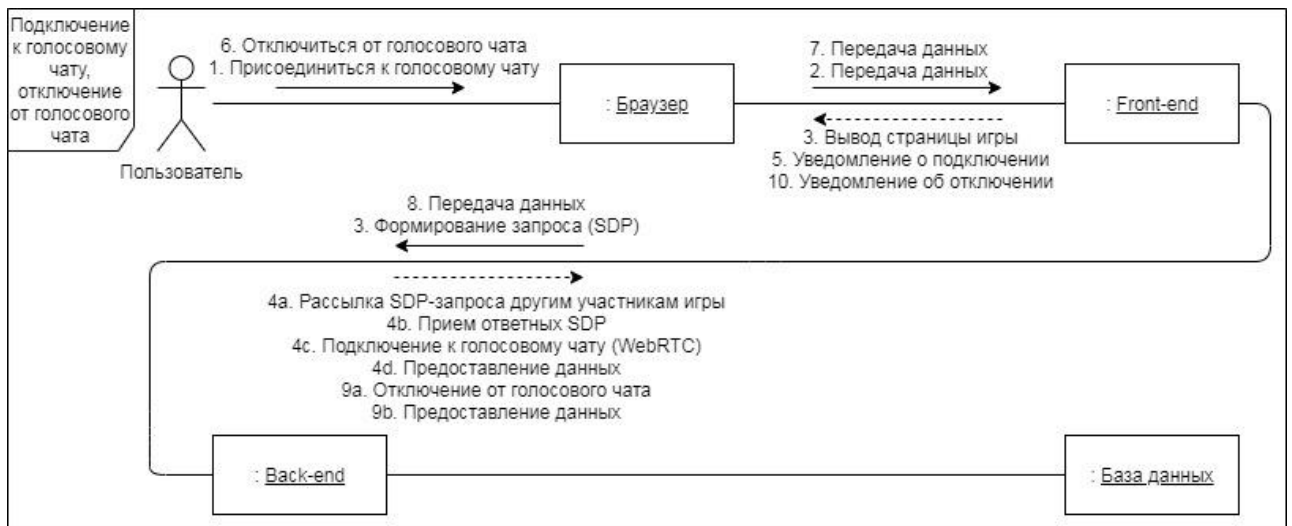


Рис. 23. Диаграмма взаимодействий для прецедентов «Подключение к голосовому чату» и «Отключение от голосового чата»

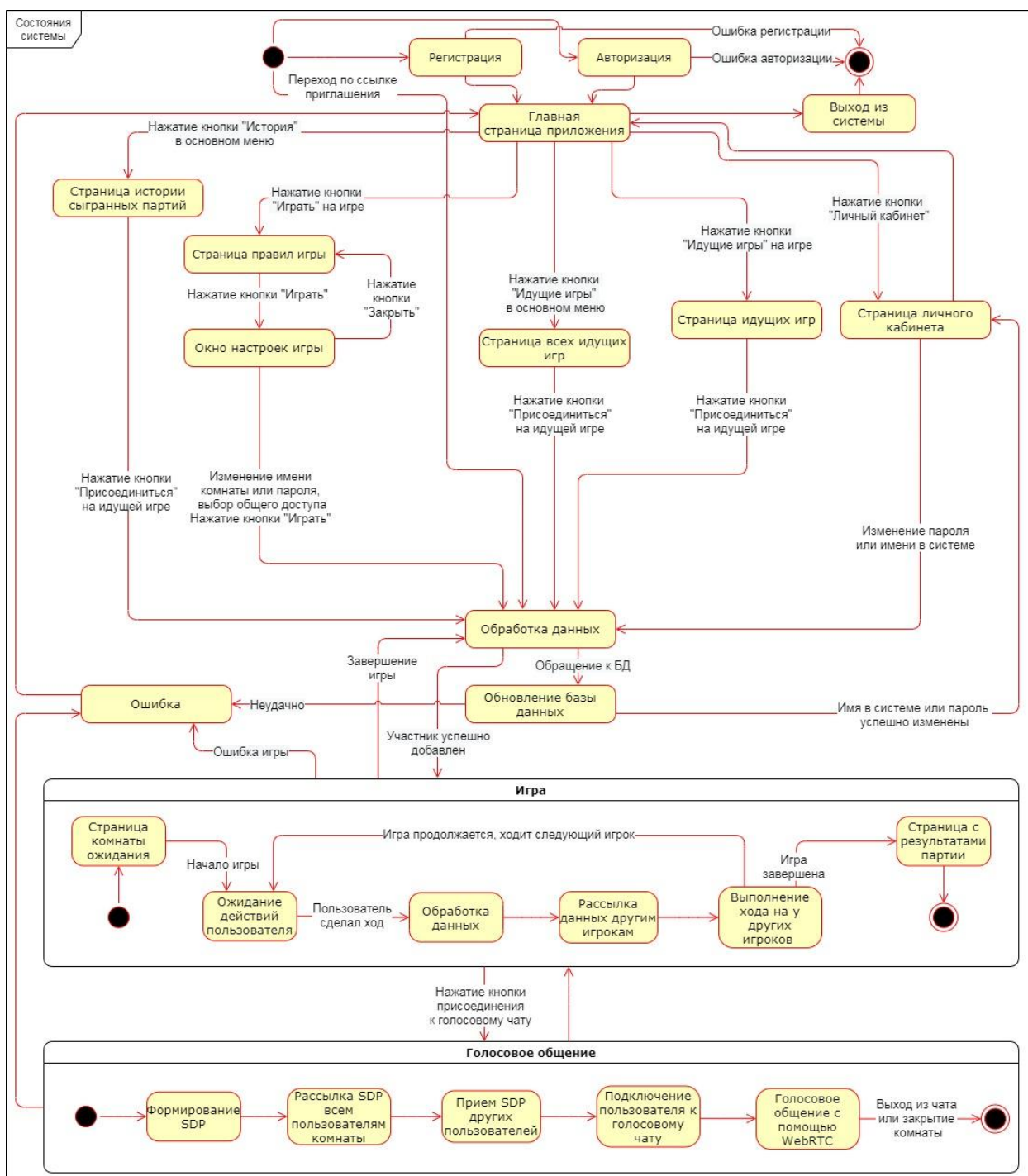
2.7.3. Варианты состояния системы

Диаграмма состояний, изображенная на рисунке 24, отображает возможные состояния системы. Запуск системы возможен при переходе по ссылке-приглашению на игру, в таком случае авторизации не требуется. Без использования ссылки при запуске приложение требует авторизоваться, после чего пользователь попадает на главную страницу.

После авторизации в системе пользователь может перейти на страницу идущих всех игр, страницу идущих партий одной игры, страницу истории. С этих страниц при нажатии кнопки «Присоединиться» на активной игре пользователь подключается к игровой комнате и игре.

Также пользователь может создать новую игровую комнату, для этого необходимо нажать кнопку «Играть» на игре в библиотеке, после этого пользователь попадет на страницу с историей партий этой игры и правилами игры, далее нажать кнопку «Играть», настроить комнату (можно изменить имя комнаты, выбрать общий доступ или доступ по паролю). После этого пользователь попадет на страницу игровой комнаты, на которой он может получить ссылку для приглашения участников.

Все участники игровой комнаты могут подключаться и отключаться от голосового чата между участниками комнаты. После окончания игры все игроки увидят окно с результатами.



2.7.4. Варианты действий в системе

Диаграмма активности, изображенная на рисунке 25, отражает возможные действия, описанные в пункте 3.2.3, состояния которых отображены на диаграмме состояний (рисунок 24).

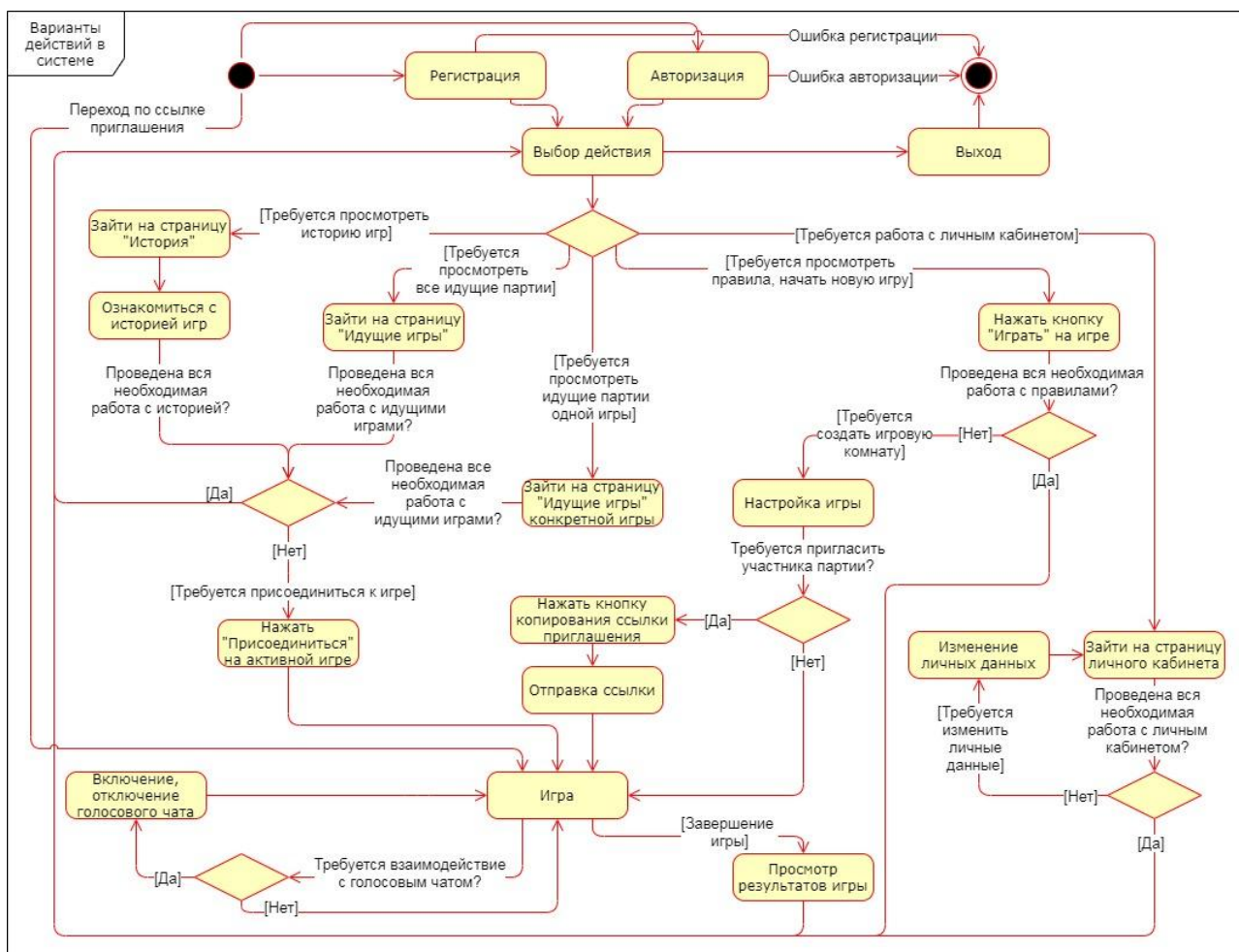


Рис. 25. Диаграмма активности

2.7.5. Развертывание приложения

На рисунке 26 представлена диаграмма развертывания, чтобы определить какие аппаратные компоненты («узлы») существуют, какие программные компоненты («артефакты») работают на каждом узле и как различные части этого комплекса соединяются друг с другом. Для разрабатываемого веб-приложения узлом устройства является компьютер, сервер и база данных, а в качестве узла среды выполнения выступает

браузер. На серверной части развернуты Front-end и Back-end части приложения и отдельно база данных.

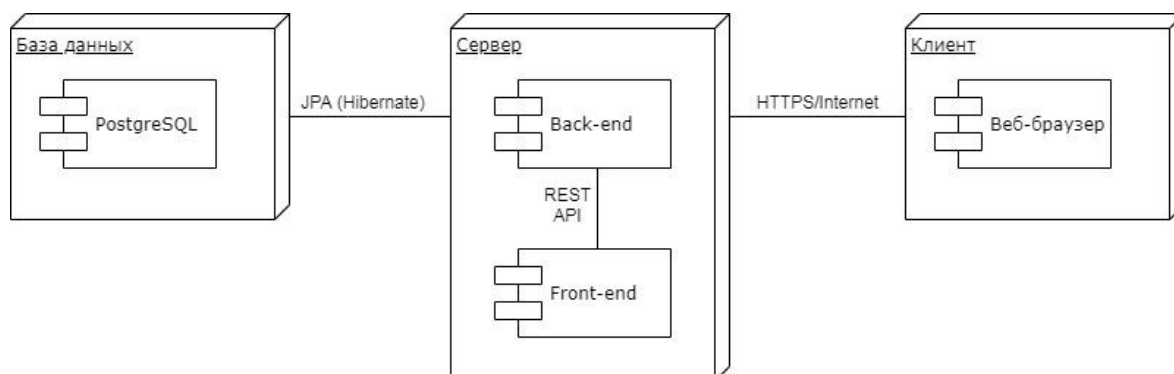


Рис. 26. Диаграмма развертывания

2.7.6. Диаграмма классов

Диаграмма классов приложения изображена на рисунке 27.

Для взаимодействия с базой данных используются классы сущностей User (пользователь), Session (игровая сессия, партия игры) и Game (игра). Для взаимодействия между игроками используется класс GameStep, содержащий информацию о ходе игрока в определенной идущей игре.

Так же используются классы-перечисления GameStatus (отображает состояние партии игры: подготавливается, идет, завершена), PrivacyStatus (доступ к комнате: общий или ограниченный), UserStatus (зарегистрирован пользователь или нет).

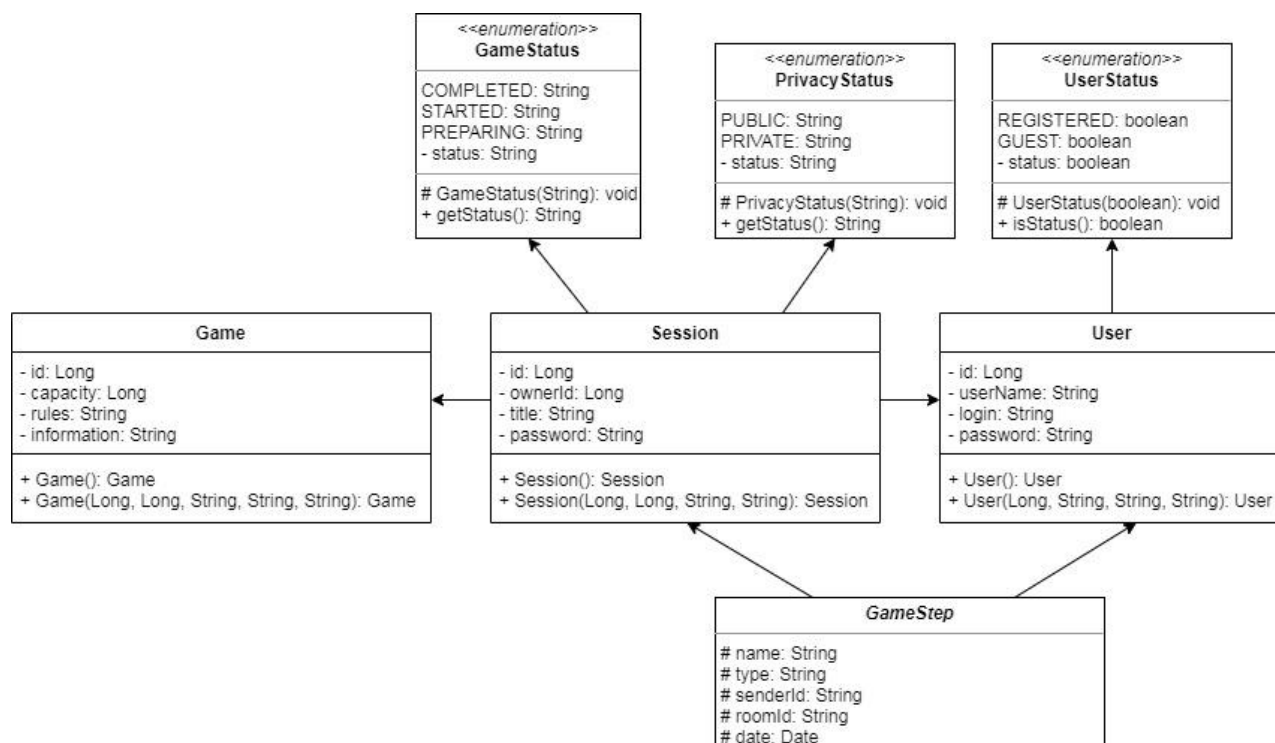


Рис. 27. Диаграмма классов

2.7.7. Диаграмма объектов

Для лучшего понимания диаграммы классов была составлена диаграмма объектов (рисунок 28). На ней отображены объекты классов сущностей в некоторый момент времени.

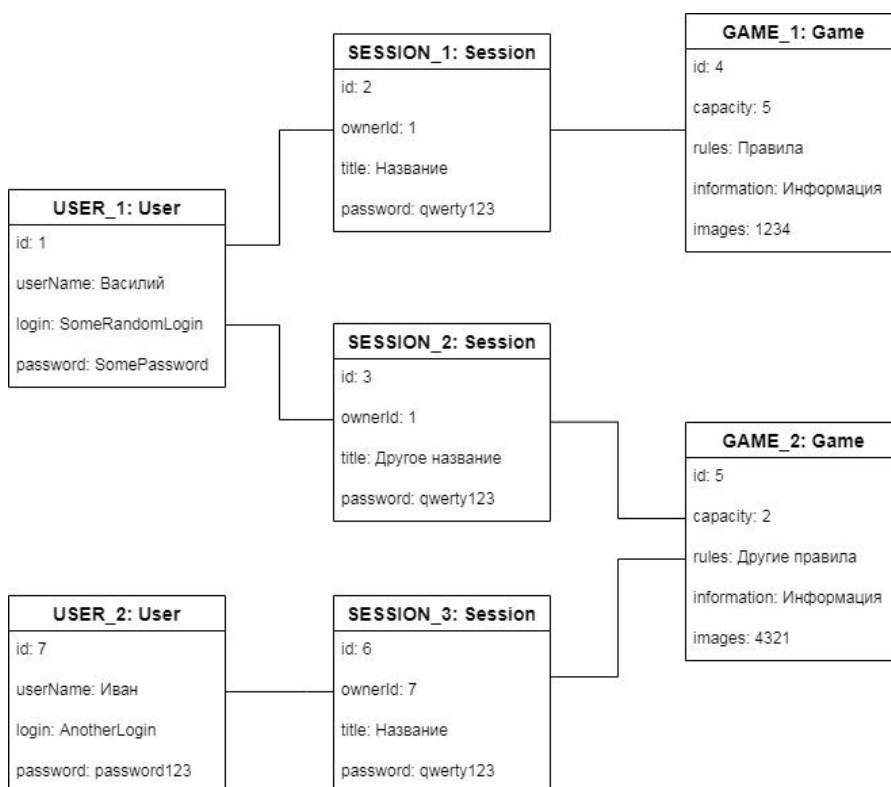


Рис. 28. Диаграмма объектов

2.7.8. IDEF0

На рисунке 29 изображена диаграмма IDEF0, иллюстрирующая работу системы. Работу сервиса регулируют Закон об авторском праве и Закон об информации. Система работает в автономном режиме. На вход в систему поступает человек, который хочет поиграть в настольные игры, на выходе система выдает человека, имеющего возможность сыграть в настольные игры в режиме онлайн с другими пользователями и статистика по действиям пользователей в системе.

Подробная расшифровка узла A0 «Работа сервиса» изображена на рисунке 30, узла A4 «Работа с системой» - на рисунке 31.



Рис. 29. IDEF0 диаграмма

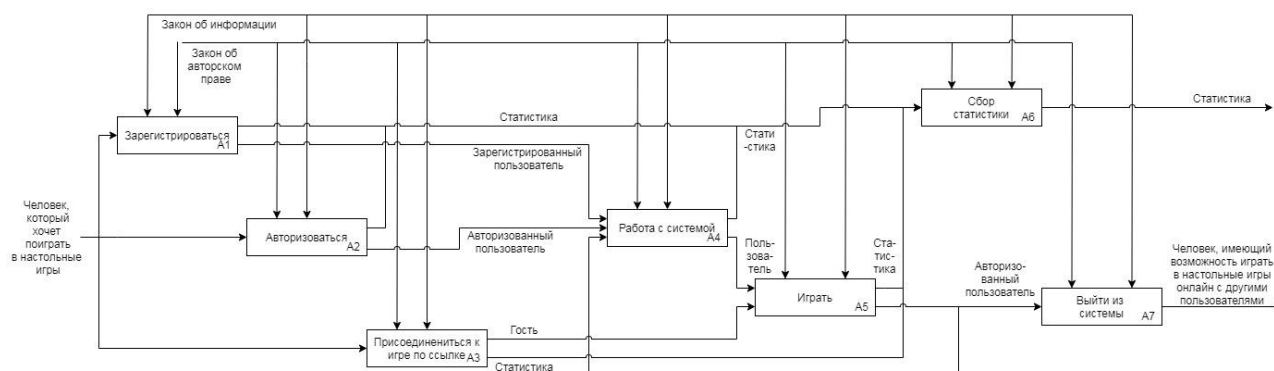


Рис. 30. IDEF0 – расшифровка узла A0 «Работа сервиса»

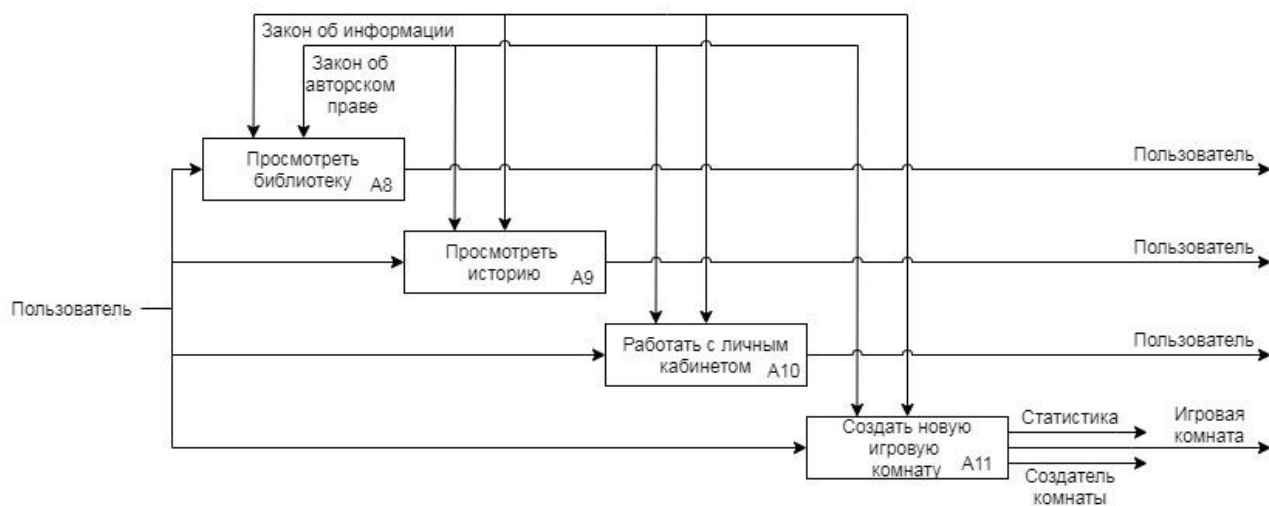


Рис. 31. IDEF0 – расшифровка узла A4 «Работа с системой»

3. Архитектура приложения

3.1 Выбор средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

Технология Spring MVC, Spring Framework с Spring Boot и Angular - продуктивный и привлекательный стек для разработки небольших веб-приложений.

Фреймворк Spring MVC обеспечивает архитектуру паттерна (Модель — Отображение (далее — Вид) — Контроллер) при помощи слабо связанных готовых компонентов. Паттерн MVC разделяет логику ввода, бизнес-логику и логику UI, обеспечивая при этом свободную связь между ними. Такая технология позволяет избегать написания многих аннотаций и конфигураций xml, и обеспечивает взаимодействие с системами Spring JDBC, Spring Security и др.

Spring Framework не только предлагает такие функции, как внедрение зависимостей или обработка транзакций, но также выступает в качестве основы для других фреймворков Spring. Лучшим примером для этого является Spring Boot. Spring Boot использует Spring Framework в качестве своей основы. Он упрощает зависимости Spring и запускает приложения прямо из командной строки. Он также не требует наличия внешнего контейнера приложений. Spring Boot помогает контролировать компоненты приложения и настраивает их извне. Благодаря таким функциям, как автоконфигурация, Spring Boot избавляет от написания лишнего кода и помогает избежать ненужной настройки.

AngularTS - это фреймворк, построенный на основе HTML и TypeScript, двух технологий, давно используемых в веб-разработке. Он позволяет использовать привычные редакторы и расширения для браузеров. Версия Angular CLI (Command Line Interface) стандартизирует структуру, позволяет создать сущности внутри приложения, а также автоматизировать его сборку.

На фреймворках разрабатываются довольно большие и сложные сайты с уникальным функционалом. Это значительно быстрее и дешевле, чем на чистом языке, но при этом такое решение позволяет разрабатывать действительно сложные вещи и оптимизировать все это под нагрузки. Кроме того, это почти всегда более безопасно, чем любая коробочная CMS.

Существует множество языков программирования, шаблонов и фреймворков. Но, благодаря многим функциям, упомянутым выше, Spring Framework и AngularTS является отличным выбором для нашего проекта.

В качестве СУБД была выбрана PostgreSQL в силу открытого доступа и высокой производительности, а также за счет поддержки данной СУБД schema-less данных, такие как JSON. PostgreSQL поддерживает интеграцию на различные платформы, взаимодействие с большинством языков программирования. PostgreSQL имеет преимущество перед другими DBMS, когда необходимо перенести базу данных из одной ОС в другую или в случае, когда реализация приложения производится на различных ОС.

Общение между Front-end и Back-end происходит средствами REST API, а обмен информации происходит с помощью передачи JSON файлов.

Для передачи информации о ходах в игре используются веб-сокеты.

Для общения пользователей используется технология WebRTC. Этот проект предназначен для организации передачи потоковых данных между браузерами по технологии точка-точка. Технология WebRTC делает возможной видео или аудиосвязь прямо через окно браузера, так что для присоединения необходимо просто перейти по ссылке на соответствующую веб-страницу. Кроме того, любые элементы интерфейса WebRTC могут быть реализованы средствами HTML5 и JavaScript/TypeScript, а используемые кодеки обеспечивают хорошее качество связи.

3.2 Архитектура системы

На рисунке 32 изображена схема архитектуры системы, на которой выделены основные компоненты.

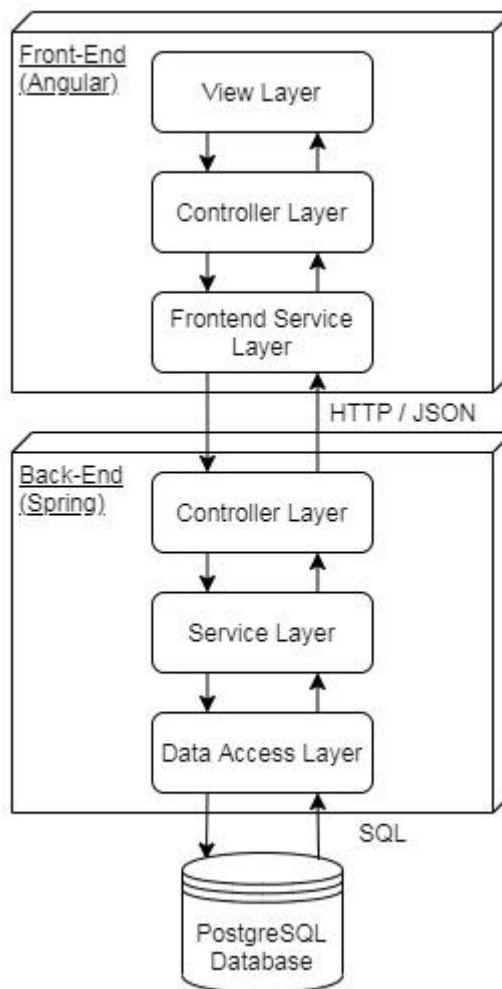


Рис. 32. Архитектура системы

Рассмотрим основные слои приложения.

Слой доступа к данным (Data Access Layer) используется для доступа к базе данных и содержит следующие элементы:

- POJO (Plain Old Java Object) классы. Они в точности описывают сущности базы данных. Именно на них осуществляется маппинг данных из БД;
- репозитории. Репозиторий является один из основных понятий в Spring Data, это несколько интерфейсов, через которые осуществляется взаимодействие с базой данных и заполнение POJO объектов необходимыми данными. Для взаимодействия с БД можно использовать как SQL, так и надстройку над ним – HQL, в котором можно оперировать объектами и их полями.

Слой сервисов (Service Layer) содержит интерфейсы сервисов и их реализации, а так же мапперы:

- вся основная функциональность по обработке данных, полученных из слоя доступа к данным или записываемых в БД, реализована в классах сервисов. Каждый класс сервиса реализует интерфейс, описывающий поведение данного класса;
- мапперы – ряд интерфейсов, осуществляющие конвертацию из POJO объектов в DTO (Data Transfer Object), которые используются для обмена данными между клиентской и серверной частями системы.

Слой контроллеров (Controller Layer) содержит REST контроллеры, определяющие поведение серверной части при обращении по URL-адресу, описанному в конкретном контроллере, также описывается и метод (POST, PUT, GET,...), на который данный контроллер будет отвечать.

Аналогично серверной части построена и клиентская часть приложения. Она так же содержит DTO, на которые происходит маппинг данных, полученных от сервера в формате JSON.

Отдельно от серверной части развернута база данных PostgreSQL, взаимодействие с которой происходит в формате SQL запросов и ответов.

3.3 Структура базы данных

Схема базы данных представлена на рисунке 33.

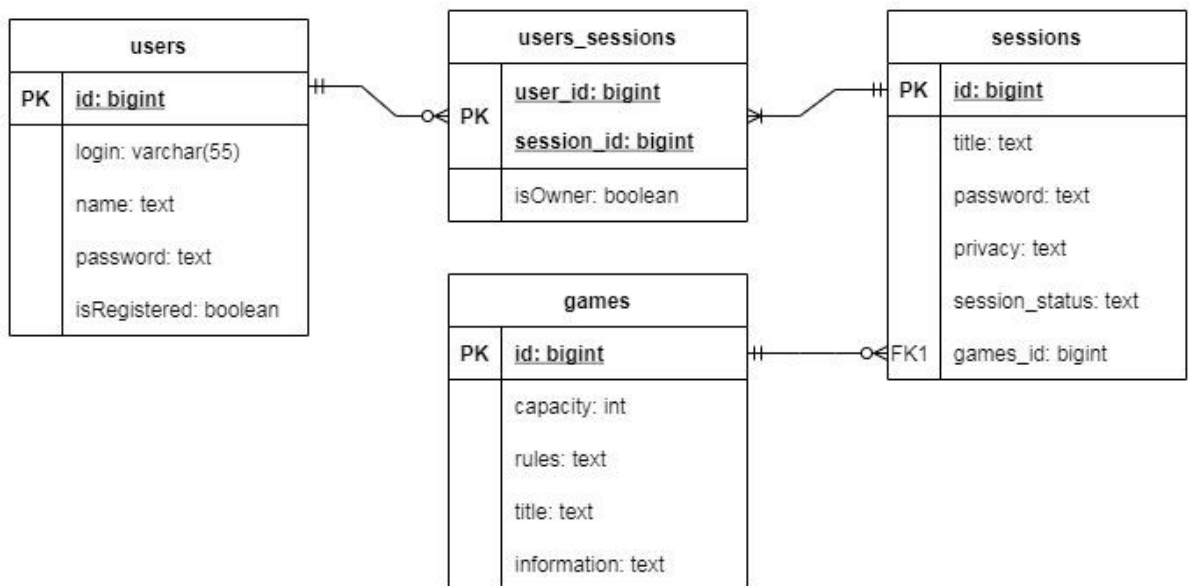


Рис. 33. Схема базы данных

В базе присутствуют 4 таблицы.

- 1) Таблица `users` используется для хранения информации о пользователе:
 - поле `id` является первичным ключом;
 - присутствуют поля для логина, имени в системе, пароля;
 - поле `isRegistered` логического типа позволяет определить, является ли пользователь зарегистрированным.
- 2) Таблица `sessions` содержит информацию об игровых сессиях (другими словами, игровых комнатах):
 - поле `id` является первичным ключом;
 - поле `title` содержит название игровой комнаты;
 - поле `password` содержит пароль от игровой комнаты;
 - поле `privacy` содержит статус приватности сессии;
 - поле `session_status` содержит статус сессии (завершена / в процессе / подготавливается);
 - поле `games_id` – внешний ключ, ссылающийся на поле `id` таблицы `games`, содержит `id` игры, сессия которой была создана (можно создать любое число сессий, связанных с конкретной игрой, сессия относится к одной и только к одной игре).

- 3) Таблица `games` содержит информацию о всех имеющихся играх:
 - поле `id` является первичным ключом;
 - имеются поля для названия игры, правил, информации и количестве игроков.
- 4) Таблица `users_sessions` используется для реализации связи многие ко многим между таблицами `users` и `sessions`. Каждый пользователь может иметь любое количество сессий, каждая сессия содержит минимум одного участника – создателя. Первичный ключ составной, содержит поля `user_id` и `session_id`, которые ссылаются на поля `id` в соответствующих таблицах.

3.4 Конфигурация приложения

Фреймворк Spring Boot предоставляет возможности автоконфигурации: автоконфигурация настраивает Spring ApplicationContext, сканируя компоненты путей к классам и регистрируя бины, соответствующие различным условиям. В разработанной системе автоконфигурация используется для подключения базе данных PostgreSQL, параметры подключения к которым заданы в файле `application.properties`: класс драйвера базы данных, ссылка на порт, на котором работает PostgreSQL, имя пользователя и пароль.

Классы репозитория помечаются аннотацией `@Repository`, классы сервисов и контроллеров - аннотациями `@Service` и `@Controller` соответственно. Эти аннотации будут использованы Spring Boot при запуске приложения.

Отдельно конфигурируются настройки Cross-Origin Resource Sharing - механизма безопасности, позволяющий веб-странице одного домена обращаться к ресурсу другого домена (кросс-доменный запрос). По умолчанию сервер запрещает запросы от других доменов, поэтому в конфигурации необходимо указать домены, с которых разрешены запросы, методы, которыми разрешено делать запросы, и внутренний URL, по

которому разрешено обращаться к серверу. Возможность конфигурации CORS предоставляет фреймворк Spring Security.

3.5 Работа с веб-сокетами

Веб-сокеты используются для обмена потоковыми данными между клиентской и серверной частями приложения, а конкретно для передачи данных о ходах в игре и подключении WebRTC.

Процесс подключения к созданной и сохраненной в базу данных игровой комнате происходит следующим образом:

- 1) при подключении к соккету от пользователь формирует запрос по протоколу SDP;
- 2) полученные данные рассылаются другим подключенным пользователям;
- 3) пользователи формируют ответы, которые пересылаются подключаемому пользователю;
- 4) таким образом настраивается соединение по сети Peer-To-Peer;
- 5) далее в процессе игры каждый ход, сделанный игроком, с помощью соккетов рассылается всем другим участникам.

Подключение к голосовому чату происходит аналогичным образом:

- 1) при подключении к голосовому чату на стороне пользователя создается объект Peer подключения, в качестве источника данных у которого указан микрофон;
- 2) пользователь формирует запрос по протоколу SDP;
- 3) полученные данные рассылаются другим подключенным пользователям;
- 4) пользователи формируют ответный SDP, который пересылается подключаемому пользователю;
- 5) при получении ответов настраивается соединение и поток звуковых данных подключается к динамику пользователя;

б) таким образом настраивается сеть Peer-To-Peer, и в дальнейшем механизм веб-соккетов при общении не используется.

3.6 Взаимодействие между серверной и клиентской частями системы. Swagger

Взаимодействие между клиентской и серверной частью системы происходит в формате REST.

Отследить запросы и ответы на них можно с помощью инструмента Swagger. Swagger - это фреймворк для спецификации REST API. Swagger UI позволяет создать интерактивную документацию API, которая позволяет проверять вызов API прямо в браузере.

Список запросов, сформированный с помощью Swagger UI, представлен на рисунке 34.

| | | | |
|-------------------------------------|--------------------------|-----------------|-----|
| LoginController | | | ^ |
| POST | /app/login/check | checkLogin | ✓ 🔒 |
| RegisterController | | | ^ |
| POST | /app/register | register | ✓ 🔒 |
| UpdatePersonalDataController | | | ^ |
| PUT | /app/update/name | updateName | ✓ 🔒 |
| PUT | /app/update/password | updatePassword | ✓ 🔒 |
| GeneratedUserController | | | ^ |
| POST | /app/generate/user | checkLogin | ✓ 🔒 |
| DeleteUserController | | | ^ |
| DELETE | /app/user/delete | userDelete | ✓ 🔒 |
| GamesController | | | ^ |
| GET | /app/activeSessions/{id} | activeGamesById | ✓ 🔒 |
| GET | /app/activeSessions | activeGames | ✓ 🔒 |
| GET | /app/sessions/{login} | sessionsByLogin | ✓ 🔒 |
| GET | /app/games | games | ✓ 🔒 |

Рис. 34. Swagger UI

4. Интерфейс приложения

4.1 Основное и дополнительное меню

На всех страницах приложения в верхней части окна отображается основное навигационное меню. С его помощью пользователь может перейти на страницу библиотеки игр (главную страницу), страницу истории сыгранных партий, страницу всех идущих игр, а также выйти из системы (если пользователь авторизован).



Рис. 35. Основное навигационное меню авторизованного пользователя




Рис. 36. Основное навигационное меню неавторизованного пользователя

Дополнительное навигационное меню содержит помеченные пользователем игры и кнопку для перехода в личный кабинет. К дополнительному меню имеют доступ только авторизованные пользователи.

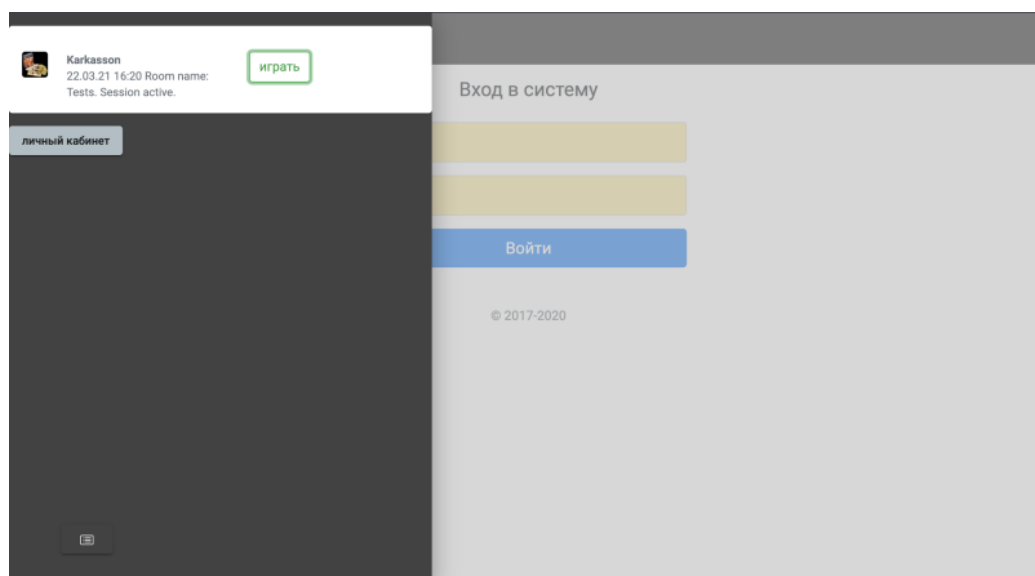


Рис. 37. Боковое (дополнительное) навигационное меню

4.2 Страница регистрации

На странице регистрации представлены следующие поля для заполнения пользователем:

- логин;
- пароль (2 раза);
- имя в системе.

Чтобы успешно зарегистрироваться в системе, пользователю необходимо заполнить указанные поля и нажать кнопку регистрации.

Библиотека Идущие игры Вход

Регистрация

Логин

Пароль

Повторите пароль

Имя в системе

Войти

© 2017-2021

Рис. 38. Страница регистрации

4.3 Страница авторизации

Для авторизации в системе пользователю необходимо указать свой логин и пароль и нажать кнопку «Войти». В случае если пользователя с такими данными не существует, будет показано сообщение об ошибке. При нажатии кнопки «Перейти к регистрации» пользователь будет перенаправлен на страницу регистрации.

Библиотека Идущие игры Вход

Вход в систему

Логин
test6

Пароль

Войти

перейти к регистрации

© 2017-2021

Рис. 39. Страница авторизации

4.4 Главная страница приложения

При успешной регистрации или авторизации пользователь направляется на главную страницу приложения, которая содержит библиотеку игр. На ней пользователь может выбрать одну из представленных игр, просмотреть все идущие партии этой игры (кнопка «Find games»), правила игры, результаты предыдущих партий или создать новую игровую комнату (кнопка «Play»).

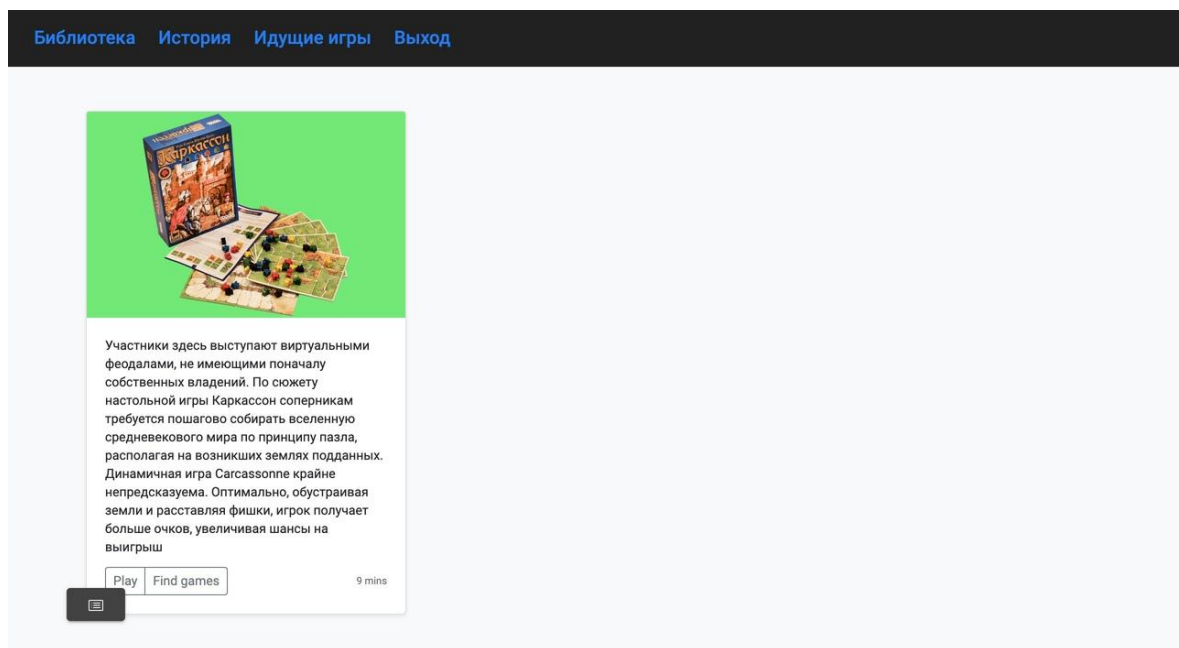


Рис. 40. Главная страница приложения

4.5 Страница игры

При нажатии кнопки «Play» на одной из игр загружается страница выбранной игры. Она содержит название игры, правила, результаты предыдущих партий (статус игры) и кнопку создания новой игровой комнаты.

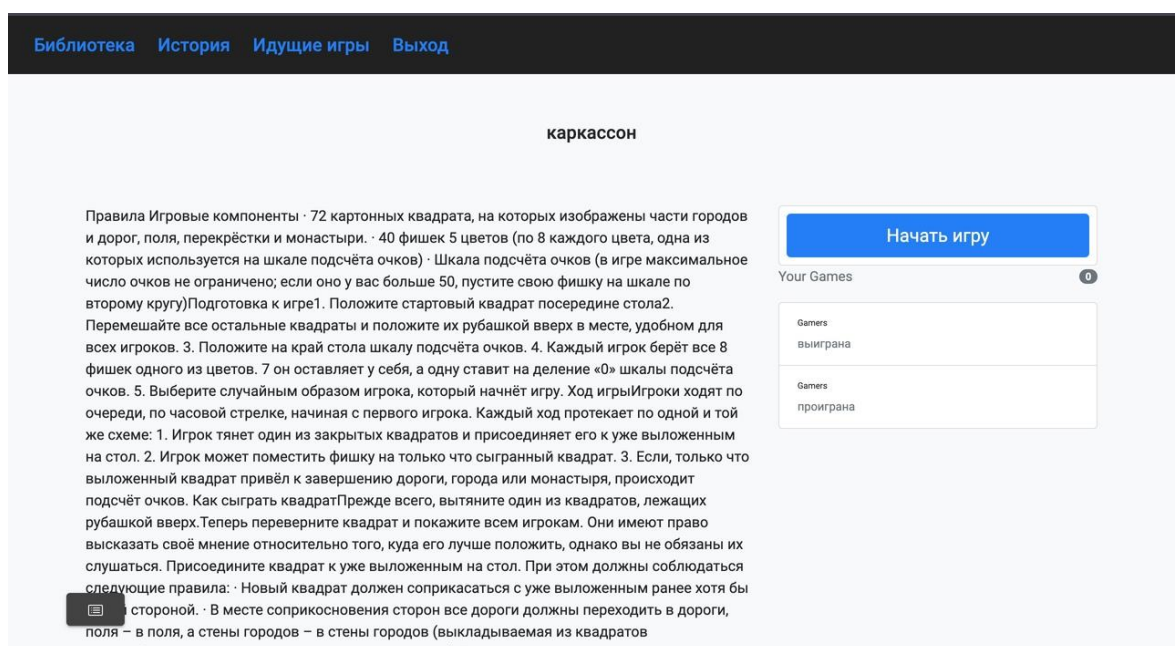


Рис. 41. Страница игры

4.6 Окно настроек игровой комнаты

При нажатии кнопки «Начать игру» загружается окно с настройками создаваемой игровой комнаты. Окно содержит:

- ссылку на комнату (ее можно будет получить и в дальнейшем);
- чекбоксы, используемые для настройки доступа (доступ по паролю или же общий доступ);
- поля для заполнения (имя комнаты и пароль, который будет запрашиваться при подключении других пользователей к игровой комнате, в случае, если выбрана опция «использовать пароль»);
- кнопки для отмены или подтверждения.

Рис. 42. Окно настроек игровой комнаты

4.7 Страница игровой комнаты

При успешном создании игровой комнаты или подключении к уже существующей, пользователь перенаправляется на страницу игровой комнаты. Она содержит:

- название игровой комнаты;

- кнопки, с помощью которых создатель комнаты может скопировать пароль и ссылку-приглашение для подключения новых игроков;
- кнопку для присоединения к голосовому чату либо для отсоединения от него;
- кнопку для подтверждения готовности начать игру;
- кнопку, при нажатии на которую начинается игровая сессия.

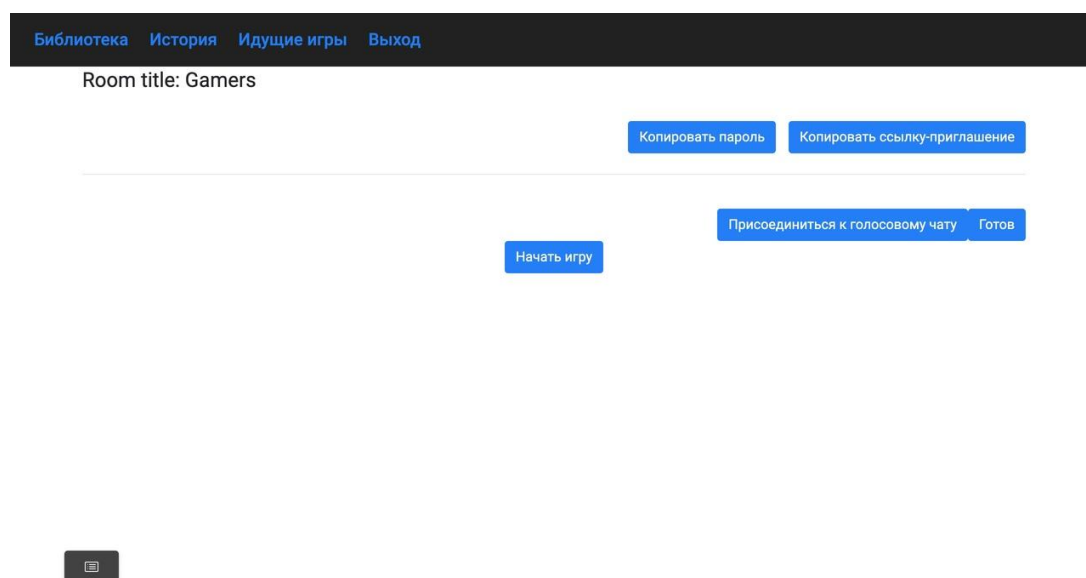


Рис. 43. Страница игровой комнаты

4.8 Страница с игровым полем

При подключении к игре пользователь попадает на страницу с самой игрой. Кроме игрового поля с необходимыми элементами, страница содержит также и навигационные меню, поскольку переход пользователя на какую-либо другую страницу системы не останавливает игровую сессию.

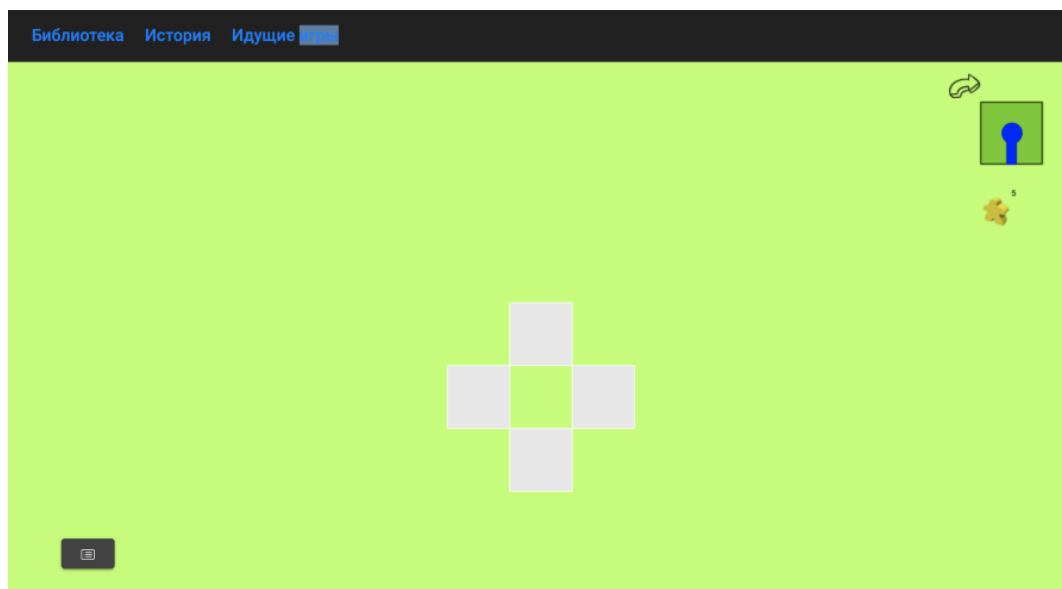


Рис. 44. Страница с игровым полем

4.9 Страница личного кабинета

При нажатии кнопки «Личный кабинет» в боковом навигационном меню пользователь направляется на страницу личного кабинета. Страница содержит:

- поля с логином и именем в системе;
- поля для смены имени;
- поля для смены пароля.

 The image shows a web interface for a user profile page. At the top is a dark navigation bar with links: "Библиотека", "История", "Идущие игры", and "Выход". Below the navigation bar is a purple banner with the text "Добро пожаловать в личный кабинет! Мое имя в системе". Under the banner are two input fields: "Мой логин" and "Мое имя в системе". Below these are two sections for changing information. The first section has input fields for "Новое имя" and "Пароль", followed by a "СМЕНИТЬ" button. The second section has input fields for "Новый пароль", "Повторите пароль", and "Пароль", followed by a "СМЕНИТЬ" button. In the bottom left corner, there is a small dark gray button with a white icon.

Рис. 45. Страница личного кабинета

4.10 Страница идущих игр

При нажатии кнопки «Идущие игры» в основном меню пользователь попадает на страницу, содержащую информацию о всех идущих партиях всех игр, находящихся в общем доступе, с возможностью присоединиться к игровой комнате.

Если пользователь перешел на страницу идущих игр, нажав кнопку «Find games» на конкретной игре, то показаны будут только идущие партии выбранной игры.

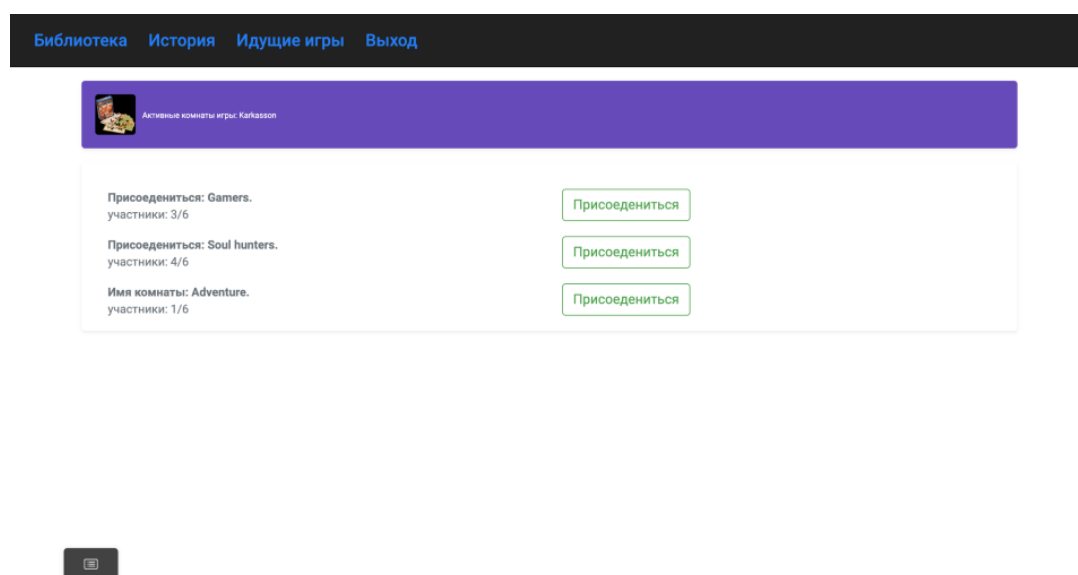









Рис. 46. Страница идущих игр

4.11 Страница истории

На станицу истории пользователь может перейти, выбрав соответствующий пункт в навигационном меню. Страница истории содержит информацию по сыгранным или идущим партиям: название игры, дата, название игровой комнаты, статус игровой сессии, счет и место (в случае, если партия завершена), кнопку для присоединения к игре (если партия активна).

My game history

Недавние игры

| | | |
|---|---|----------------|
|  | Karkasson 22.03.21 16:45 Имя комнаты: Gamers. Session: active. | присоединиться |
|  | Karkasson 22.03.21 16:20 Имя комнаты: Tests. Session active. | присоединиться |
|  | Karkasson 22.03.21 14:30 Имя комнаты: Happiness. Session active. | присоединиться |
|  | Karkasson 21.03.21 18:26 Имя комнаты: Gamers. Session ended. Ваш счет: 78. Место 3. | |
|  | Karkasson 19.03.21 19:44 Имя комнаты: Gamers. Session ended. Ваш счет: 128. Место 2. | |
|  | Karkasson 19.03.21 13:25 Имя комнаты: Gamers. Session ended. Ваш счет: 136. Место 1. | |
|  | Karkasson 18.03.21 19:02 Имя комнаты: Gamers. Session ended. Ваш счет: 71. Место 4. | |




Рис. 47. Страница истории

5. Тестирование

После реализации всех задач, был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- дымовое тестирование;
- UI тестирование;
- usability тестирование.

5.1 Дымовое тестирование

В ходе данного тестирования проверялась работоспособность приложения на следующих основных сценариях:

1. регистрация пользователя;
2. авторизация пользователя;
3. изменение личных данных пользователя;
4. выбор игры из библиотеки игр;
5. просмотр идущих игр;
6. присоединение к идущей игре с помощью интерфейса;
7. присоединение к идущей игре по ссылке;
8. создание новой игровой комнаты;
9. общение с помощью голосового чата;
10. участие в игре и просмотр результатов партии.

Дымовое тестирование проводилось ручным способом, на развернутом на сервере и открытом в браузере Opera версии 76.0.4009.0 приложении, с активным соединением с сетью интернет для связи с Back-end частью приложения.

Результаты, полученные в ходе тестирования, представлены в таблице

2.

Таблица 2. – Результаты дымового тестирования

| Сценарий | Результат |
|--|------------------|
| Регистрация пользователя | Пройден |
| Авторизация пользователя | Пройден |
| Изменение личных данных пользователя | Пройден |
| Выбор игры из библиотеки игр | Пройден |
| Просмотр идущих игр | Пройден |
| Присоединение к идущей игре с помощью интерфейса | Пройден |
| Присоединение к идущей игре по ссылке | Пройден |
| Создание новой игровой комнаты | Пройден |
| Общение с помощью голосового чата | Пройден |
| Участие в партии какой-либо игры и просмотр результатов партии | Пройден |

По итогам дымового тестирования было установлено, что приложение проходит все основные сценарии.

5.2 UI тестирование

В ходе UI тестирования было выполнено 24 теста, охватывающие почти все возможности приложения. Результаты, полученные в ходе тестирования, представлены в таблице 3.

Таблица 3. – Результаты UI тестирования

| № | Сценарий | Шаги | Ожидаемый результат | Статус |
|----------|------------------------------------|--|--|---------------|
| 1 | Авторизация | 1. Открывается страница входа в приложение; 2. В поле «логин» вводится «user»; 3. В поле «пароль» вводится «password»; 4. Нажимается кнопка «Войти». | Загрузка главной страницы приложения | Пройден |
| 2 | Авторизация (неверный пароль) | Аналогично пункту 1, но на шаге 3 вводится «another» | Появление сообщения об ошибке | Пройден |
| 3 | Регистрация | 1. Открывается страница регистрации; 2. В поле «логин» вводится «login»; 3. В поле «пароль» вводится «password»; 4. В поле «повторение пароля» вводится «password» 5. В поле «имя» вводится «name»; 6. Нажимается кнопка «Войти». | Загрузка главной страницы приложения | Пройден |
| 4 | Регистрация (несовпадающие пароли) | Аналогично пункту 3, но на шаге 4 вводится «another» | Появление сообщения о несовпадении паролей | Пройден |
| 5 | Просмотр личного кабинета | 1. В боковом меню нажимается кнопка «Личный кабинет» | Загрузка страницы личного кабинета | Пройден |
| 6 | Изменение имени в системе | 1. Открывается страница личного кабинета; 2. В поле «новое имя» вводится «New name»; 3. В поле «пароль» вводится «password»; 4. Нажимается кнопка «сменить». | В поле «мое имя в системе» в верхней части страницы значение изменяется с «name» на «New name» | Пройден |

Продолжение таблицы 3.

| | | | | |
|----|--|---|---|---------|
| 7 | Изменение имени в системе (неверный пароль) | Аналогично 6, но на 3 шаге вводится «another» | Появление сообщения о неверном пароле | Пройден |
| 8 | Изменение пароля | 1. Открывается страница личного кабинета; 2. В поле «новый пароль» вводится «new_password»; 3. В поле «повторите пароль» вводится «new_password»; 4. В поле «пароль» вводится «password»; 5. Нажимается кнопка «сменить». | Изменение пароля (проверяется с помощью выхода из системы и авторизации по новому паролю) | Пройден |
| 9 | Изменение пароля (несовпадающие пароли) | Аналогично 8, но на шаге 3 вводится «another» | Появление сообщения о несовпадении паролей | Пройден |
| 10 | Изменение пароля (неверный пароль) | Аналогично 8, но на шаге 4 вводится «another» | Появление сообщения о неверном пароле | Пройден |
| 11 | Выбор игры из библиотеки игр, просмотр правил игры и статистики партий в этой игре | 1. Загружается главная страница приложения; 2. На карточке игры нажимается кнопка «Play». | Загрузка страницы игры, содержащей правила и статистику сыгранных партий | Пройден |
| 12 | Просмотр всех идущих игр | 1. Нажимается кнопка «Идущие игры» в меню. | Загрузка страницы идущих игр | Пройден |
| 13 | Просмотр идущих партий одной игры | 1. На карточке игры нажимается кнопка «Find games». | Загрузка страницы идущих | Пройден |

Продолжение таблицы 3.

| | | | | |
|----|--|---|--|---------|
| 14 | Присоединение к игровой сессии (UI, общий доступ) | 1. Загружается страница идущих игр; 2. На активной игре нажимается кнопка «Присоединиться». | Загрузка страницы игровой комнаты; | Пройден |
| 15 | Присоединение к игровой сессии (UI, доступ по паролю) | 1-2. Аналогично пункту 14; 3. В поле «пароль» вводится «password». | Загрузка страницы игровой комнаты | Пройден |
| 16 | Присоединение к игровой сессии (UI, доступ по паролю, неверный пароль) | 1-2. Аналогично пункту 14; 3. В поле «пароль» вводится «another». | Появление сообщения о неверном пароле | Пройден |
| 17 | Присоединение к игровой сессии (ссылка) | 1. Переход по ссылке-приглашению. | Загрузка страницы игровой комнаты | Пройден |
| 18 | Настройка и создание новой игровой комнаты (общий доступ) | 1. Загружается страница игры; 2. Нажимается кнопка «Начать игру»; 3. В поле «Имя комнаты» вводится «goom»; 4. Выбирается опция «Общий доступ»; 5. Нажимается кнопка «Начать». | Загрузка страницы игровой комнаты (общий доступ проверяется при присоединении к комнате) | Пройден |

Продолжение таблицы 3.

| | | | | |
|----|---|---|--|---------|
| 19 | Настройка и создание новой игровой комнаты (доступ по паролю) | 1-3. Аналогично пункту 18; 4. Выбирается опция «Использовать пароль»; 5. В поле «пароль» вводится «password». 6. Нажимается кнопка «Начать». | Загрузка страницы игровой комнаты (доступ по паролю проверяется при присоединении к комнате) | Пройден |
| 20 | Получение ссылки для приглашения | 1. Загружается страница игровой комнаты; 2. Нажимается кнопка «копировать ссылку». | Копирование в буфер обмена действующей ссылки на текущую игровую комнату | Пройден |
| 21 | Получение пароля от игровой комнаты | 1. Загружается страница игровой комнаты; 2. Нажимается кнопка «копировать пароль» | Копирование в буфер обмена действующего пароля от текущей игровой комнаты | Пройден |
| 22 | Подключение к голосовому чату | 1. Загружается страница игровой комнаты; 2. Нажимается кнопка «присоединиться к голосовому чату». | Присоединение к голосовому чату | Пройден |
| 23 | Участие в игре | 1. Загружается страница игровой комнаты; 2. Нажимается кнопка «Начать»; 3. Совершается какое-либо действие на игровом поле. | 1. Загрузка страницы с игровым полем; 2. Изменение объектов на игровом поле пользователем; 3. Изменение объектов на игровом поле другими пользователями. | Пройден |
| 24 | Просмотр истории | 1. Нажимается кнопка «История» в навигационном меню | Загрузка страницы истории | Пройден |

5.3 Usability тестирование

Для проведения Usability тестирования были отобраны 5 человек, ранее не пользовавшиеся приложением. В рамках данного тестирования необходимо было проверить следующие основные сценарии взаимодействия пользователей с приложением:

- регистрация;
- авторизация;
- изменение данных в личном кабинете;
- выбор игры из библиотеки;
- просмотр идущих игр;
- присоединение к идущей игре с помощью интерфейса;
- присоединение к идущей игре с помощью ссылки;
- создание игровой комнаты;
- подключение к голосовому чату;
- участие в игре.

Результаты тестирования представлены в таблице 4.

Таблица 4. – Результаты Usability тестирования

| Сценарий | Польз.1 | Польз.2 | Польз.3 | Польз.4 | Польз.5 |
|-------------------------------------|---------|---------|---------|---------|---------|
| Регистрация | Пройден | Пройден | Пройден | Пройден | Пройден |
| Авторизация | Пройден | Пройден | Пройден | Пройден | Пройден |
| Изменение данных | Пройден | Пройден | Пройден | Пройден | Пройден |
| Выбор игры | Пройден | Пройден | Пройден | Пройден | Пройден |
| Просмотр идущих игр | Пройден | Пройден | Пройден | Пройден | Пройден |
| Присоединение к игре (интерфейс) | Пройден | Пройден | Пройден | Пройден | Пройден |

Продолжение таблицы 4.

| | | | | | |
|-----------------------------------|---------|---------|---------|---------|---------|
| Присоединение к игре (ссылка) | Пройден | Пройден | Пройден | Пройден | Пройден |
| Создание новой игровой комнаты | Пройден | Пройден | Пройден | Пройден | Пройден |
| Голосовой чат | Пройден | Пройден | Пройден | Пройден | Пройден |
| Участие в игре | Пройден | Пройден | Пройден | Пройден | Пройден |

В результате все пользователи без ощутимых проблем смогли выполнить основные действия в системе.

Заключение

В результате работы было разработано веб-приложение, реализующее функциональность игровой платформы, специализирующейся на жанре настольных игр. Система не зависит от сторонних платформ и легко масштабируется.

В процессе работы выполнены следующие задачи:

1. проведен анализ рынка с целью выявления достоинств и недостатков схожих по функционалу систем;
2. спроектирована система с учетом информации, полученной ранее в ходе анализа;
3. разработана база данных;
4. разработана Back-end часть приложения;
5. разработана Front-end часть приложения;
6. создана связь между Front-end и Back-end частями приложения;
7. создана и развернута первая тестовая игра («Каркассон»);
8. проведен запланированный набор тестирований;
9. описан процесс разработки и результат.

Система отвечает всем заявленным требованиям.

Список использованных источников

1. Активация CORS для RESTful Web-сервиса: [Электронный ресурс]. URL: <https://spring-projects.ru/guides/rest-service-cors/> (Дата обращения: 06.04.2021).
2. Как выбрать технологию для большого и не очень большого веб-проекта: [Электронный ресурс]. URL: <https://tproger.ru/articles/which-technology-to-choose/> (Дата обращения: 26.02.2021).
3. Про Session Descriptor Protocol: [Электронный ресурс]. URL: <https://wiki.merionet.ru/ip-telephonya/50/pro-session-description-protocol/> (Дата обращения: 20.04.2021).
4. About Swagger Specification | Documentation | Swagger: [Электронный ресурс]. URL: <https://swagger.io/docs/specification/about/> (Дата обращения: 18.05.2021).
5. Accessing Data with JPA: [Электронный ресурс]. URL: <https://spring.io/guides/gs/accessing-data-jpa/> (Дата обращения: 15.04.2021).
6. AVTarasov2000/TPTableGame: [Электронный ресурс]. URL: <https://github.com/AVTarasov2000/TPTableGame> (Дата обращения: 17.06.2021).
7. Building REST services with Spring: [Электронный ресурс]. URL: <https://spring.io/guides/tutorials/rest/> (Дата обращения: 28.03.2021).
8. WebRTC: [Электронный ресурс]. URL: <https://webrtc.org/> (Дата обращения: 19.04.2021).
9. UML – краткое руководство: [Электронный ресурс]. URL: <https://coderlessons.com/tutorials/akademicheskii/uchit-uml/uml-kratkoe-rukovodstvo/> (Дата обращения: 01.05.2021).