

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	6
2.1 Целевая аудитория	6
2.2 Описание предметной области	8
2.3 Пользователи системы	9
2.4 Границы функциональности	10
2.5 Анализ существующих решений	11
2.6 Воронки конверсии	16
2.6.1. Цель «Регистрация»	17
2.6.2. Цель «Создание игровой комнаты»	17
2.6.3. Цель «Присоединение к игровой комнате по ссылке»	17
2.7 Анализ задачи	18
2.7.1. Варианты использования приложения	18
2.7.2. Взаимодействие компонентов системы	19
2.7.3. Варианты состояния системы	26
2.7.4. Варианты действий в системе	28
2.7.5. Развертывание приложения	28
2.7.6. Диаграмма классов	29
2.7.7. Диаграмма объектов	30
2.7.8. IDEF0	31
3. Архитектура приложения	33
4. Проектная часть	34
4.1 Интерфейс приложения	34
4.2 Swagger	34

5.	Тестирование.....	34
5.1	Smoke testing.....	35
5.2	UI тестирование	35
5.3	Usability тестирование.....	35
6.	Заключение.....	35

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 сайтике, 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 уникальной платформы, поддерживающей работу пошаговых настольных онлайн-игр и позволяющей пользователю окунуться в игровой мир в компании своих товарищей.

1. Постановка задачи

Проект «Настольный сервер» является веб-сервисом для настольных игр. Он предназначен для всех, кто любит играть и общаться с друзьями.

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;
- 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 в качестве активного игрока или наблюдателя;
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 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

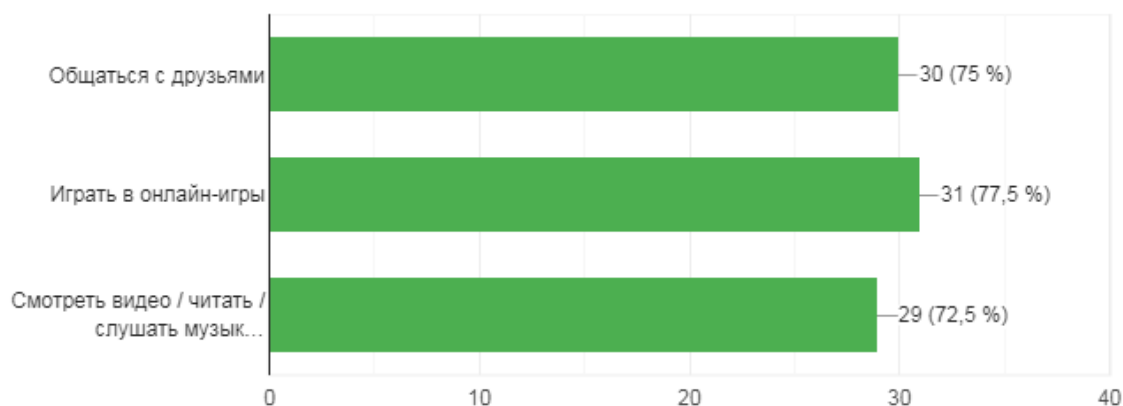


Рис. 1. Результаты опроса

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

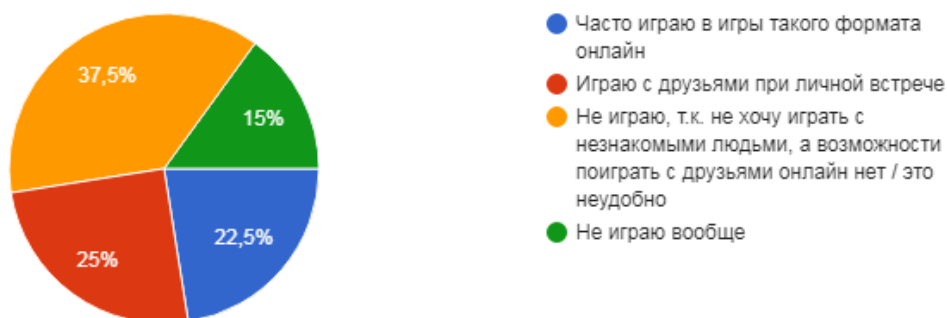


Рис. 2. Результаты опроса

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, наше 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://igroutka.net/igry-strategii/24127-karkasson.html>

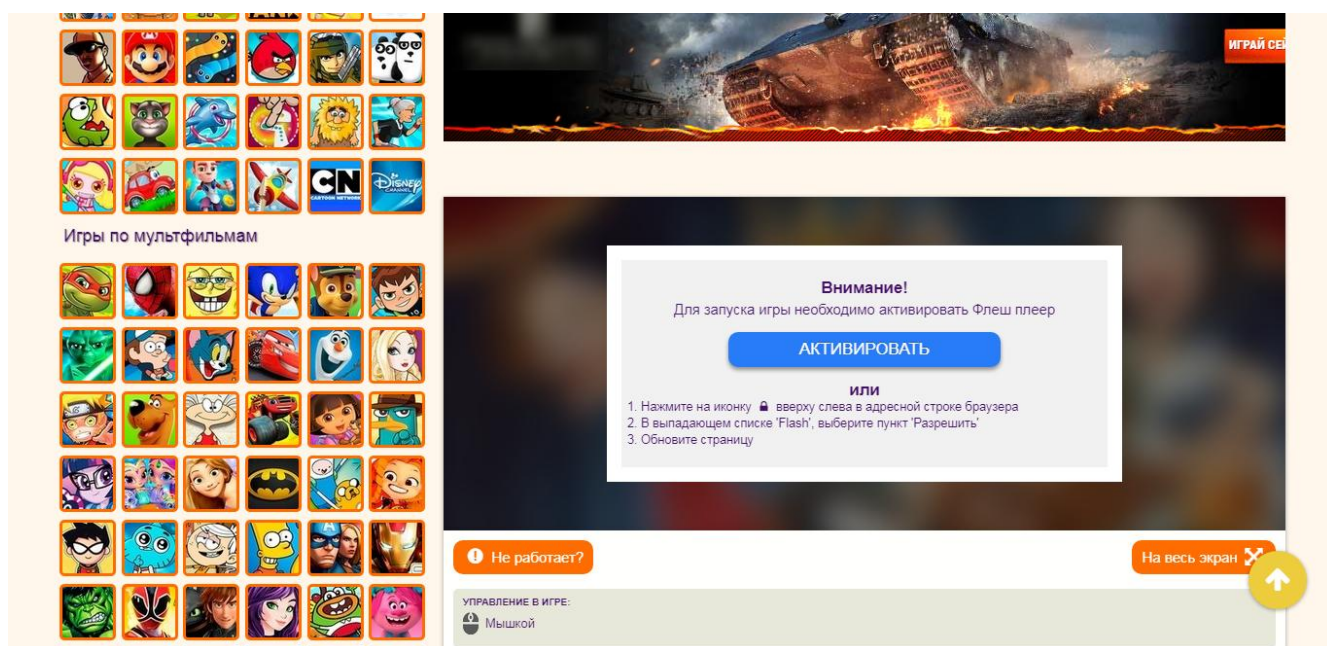


Рис. 4. Интерфейс сервиса «ИГРОУТКА»

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>

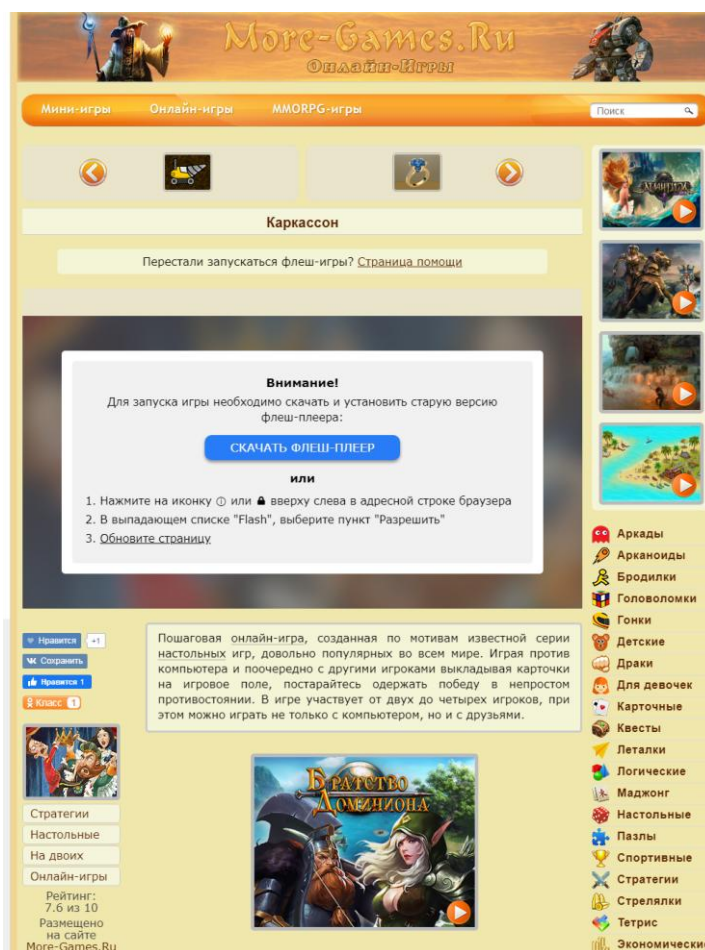


Рис. 5. Интерфейс сервиса «More-Games.Ru»

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>

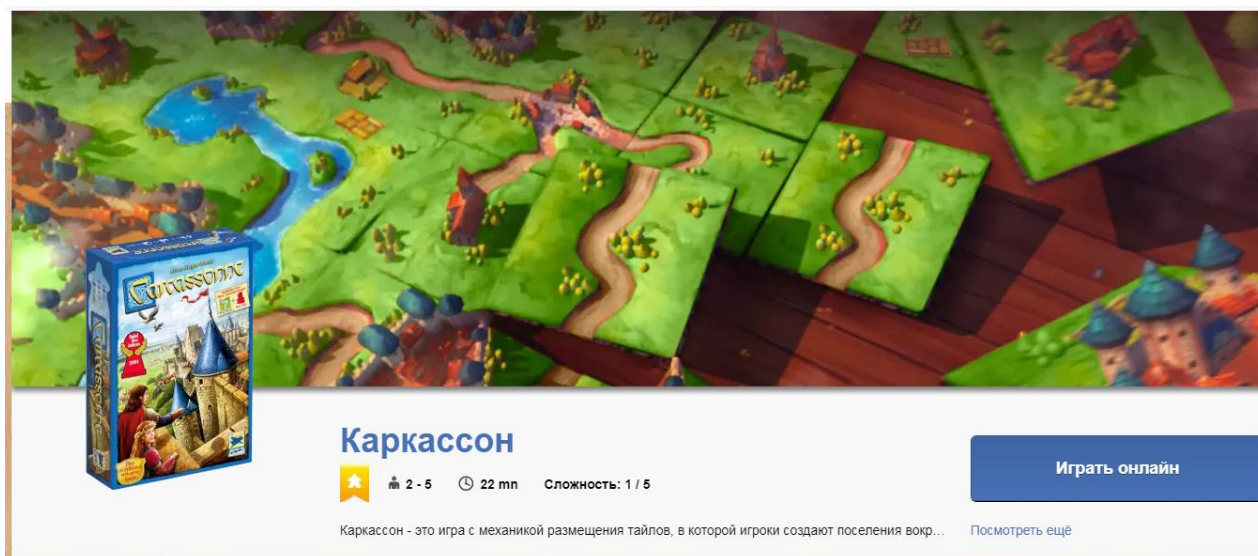


Рис. 6. Интерфейс сервиса «BoardGameArena»

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.

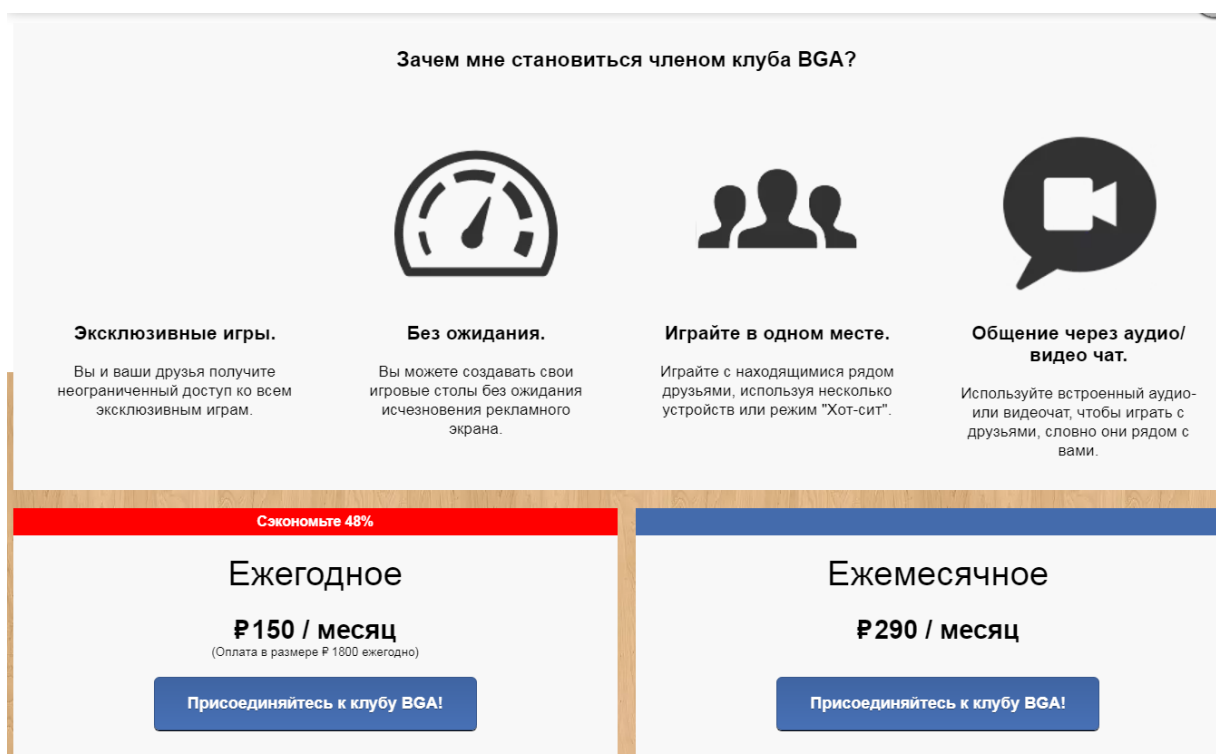


Рис. 7. Предложение покупки в приложении «BoardGameArena»

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:

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

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

Таблица 1. – Недостатки аналогов и варианты их устранения

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.

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».

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».

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.

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. 8. 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 10–15.

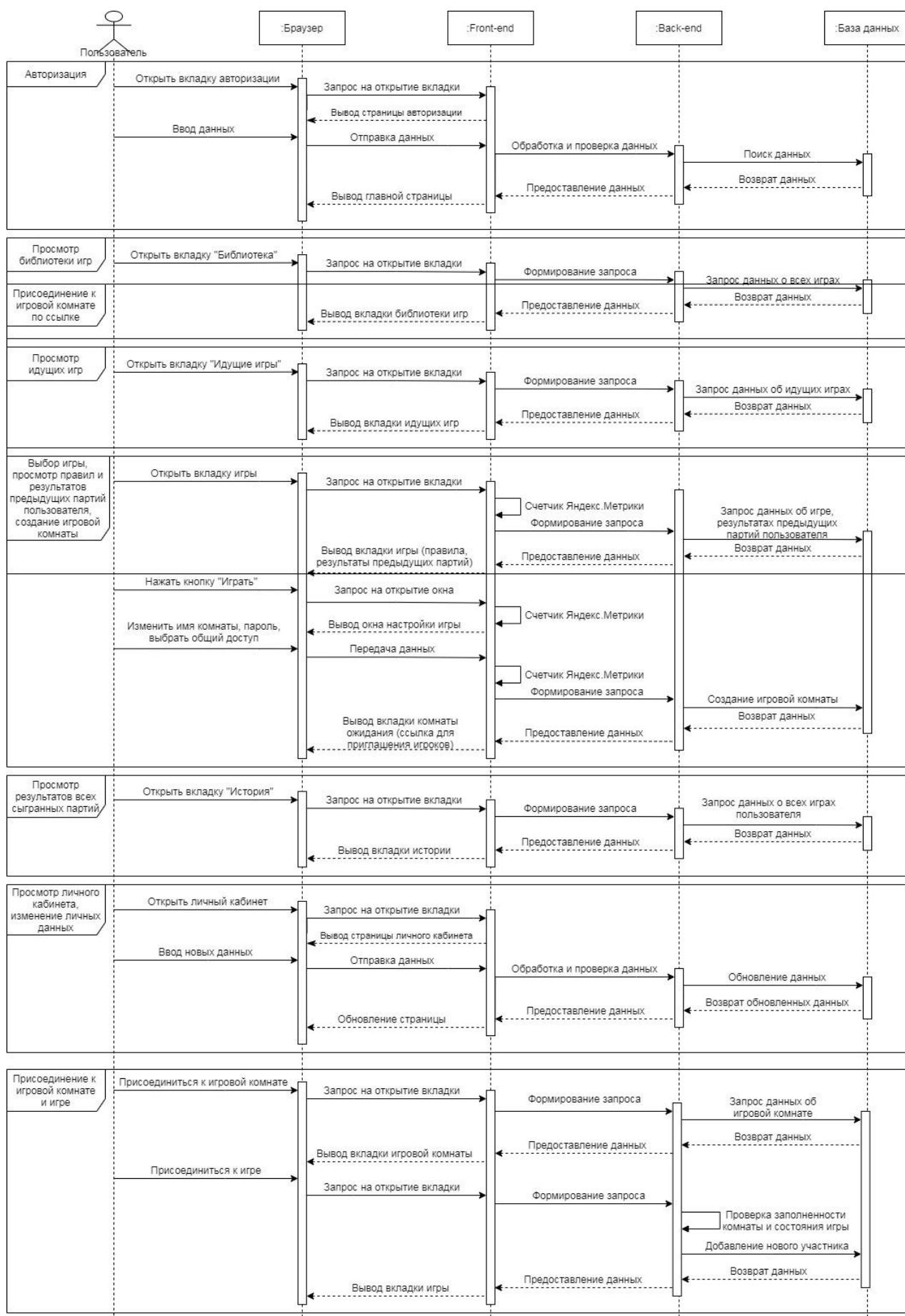


Рис. 9. Диаграмма последовательностей для пользователя



Рис. 10. Диаграмма взаимодействий для прецедента «Авторизация»



Рис. 11. Диаграмма взаимодействий для прецедента «Просмотр библиотеки» и «Начало игры»



Рис. 12. Диаграмма взаимодействий для прецедента «Просмотр идущих игр»



Рис. 13. Диаграмма взаимодействий для прецедента «Просмотр личного кабинета, изменение личных данных»



Рис. 14. Диаграмма взаимодействий для прецедента «Просмотр результатов всех сыгранных партий»

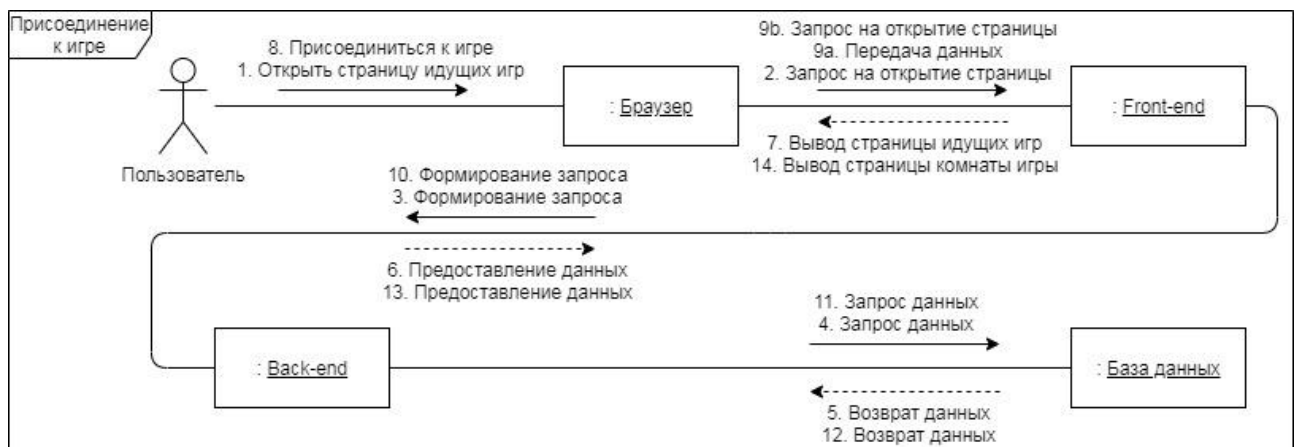


Рис. 15. Диаграмма взаимодействий для прецедента «Присоединение к игровой комнате и к игре»

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 17–18 соответственно.

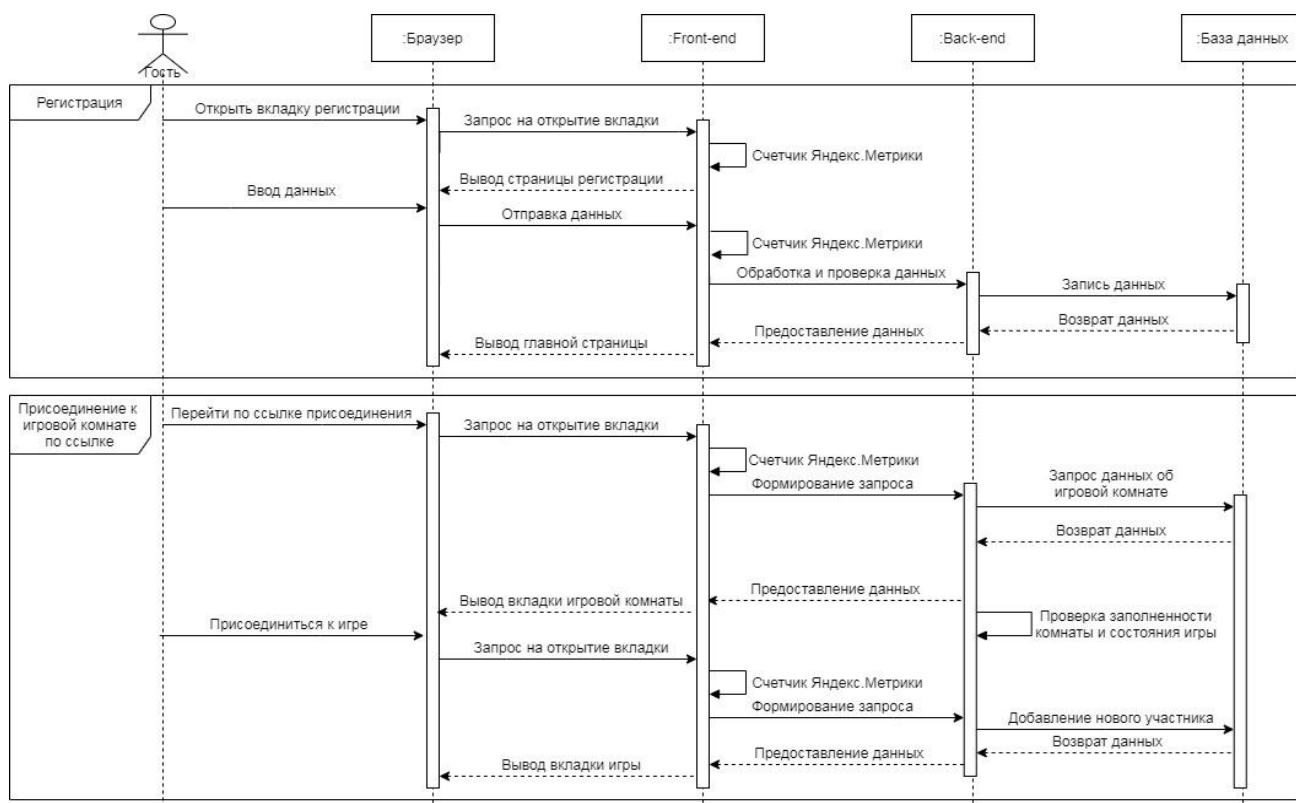


Рис. 16. Диаграмма последовательностей для гостя



Рис. 17. Диаграмма взаимодействий для прецедента «Регистрация»

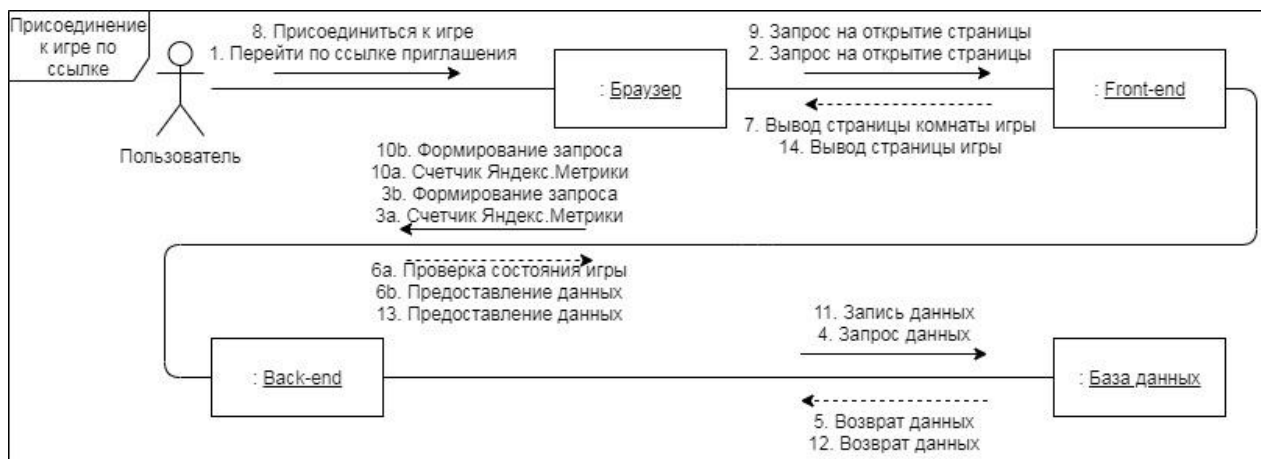


Рис. 18. Диаграмма взаимодействий для прецедента «Присоединение к игре по ссылке»

На рисунке 19 показана диаграмма последовательности 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.

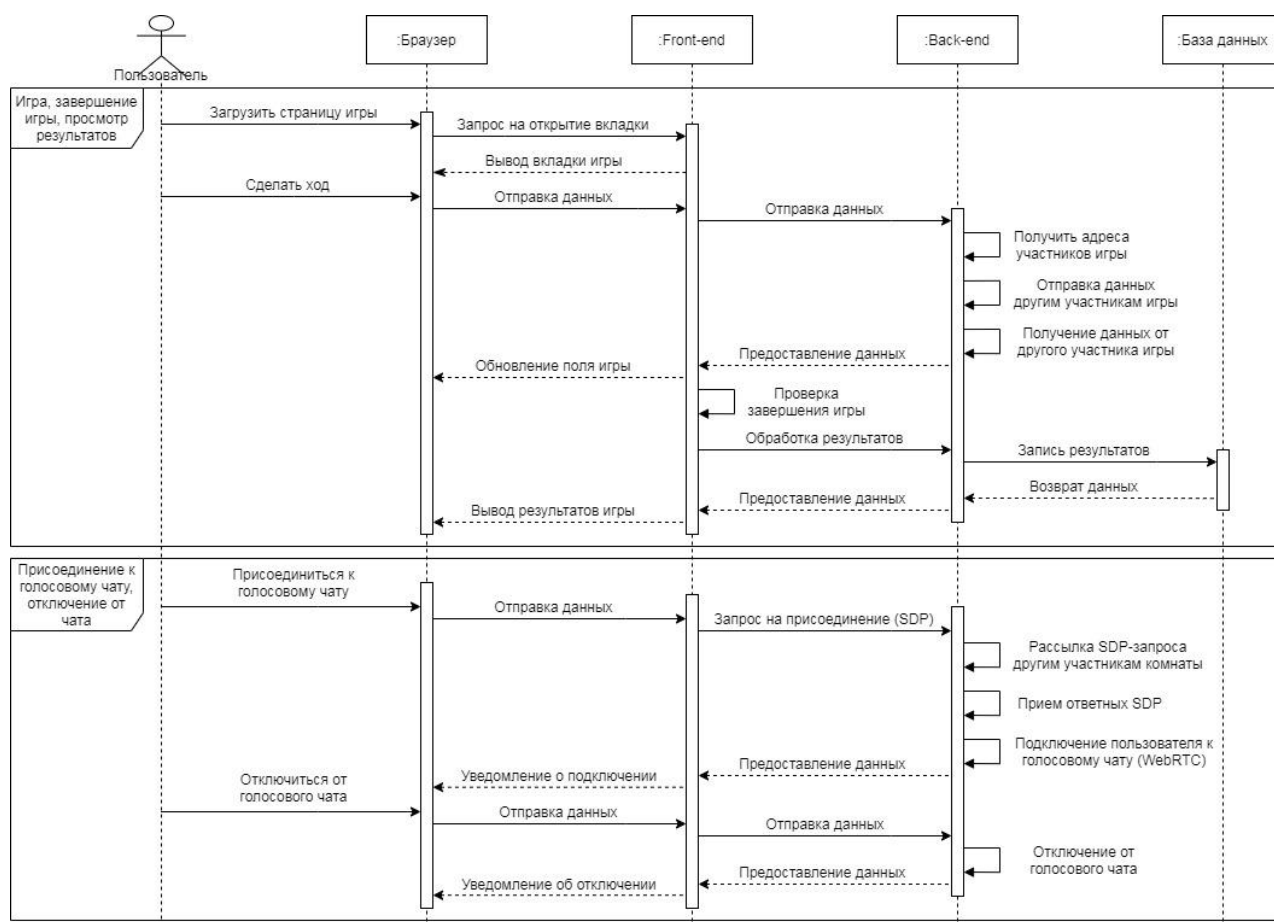


Рис. 19. Диаграмма последовательностей для взаимодействий, проходящих одинаково для пользователя и гостя

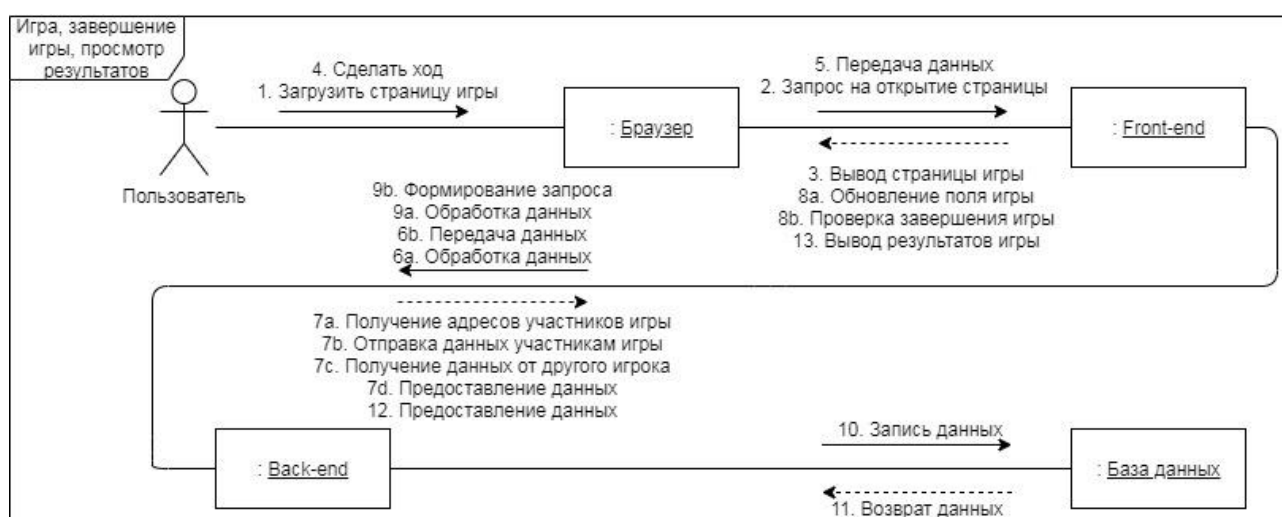


Рис. 20. Диаграмма взаимодействий для прецедентов «Игра», «Завершение игры» и «Просмотр результатов»



Рис. 21. Диаграмма взаимодействий для прецедентов «Подключение к голосовому чату» и «Отключение от голосового чата»

2.7.3. Варианты состояния системы

Диаграмма состояний, изображенная на рисунке 22, 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.

Авторизованный пользователь также может перейти на страницу личного кабинета, изменить личные данные и выйти из системы.

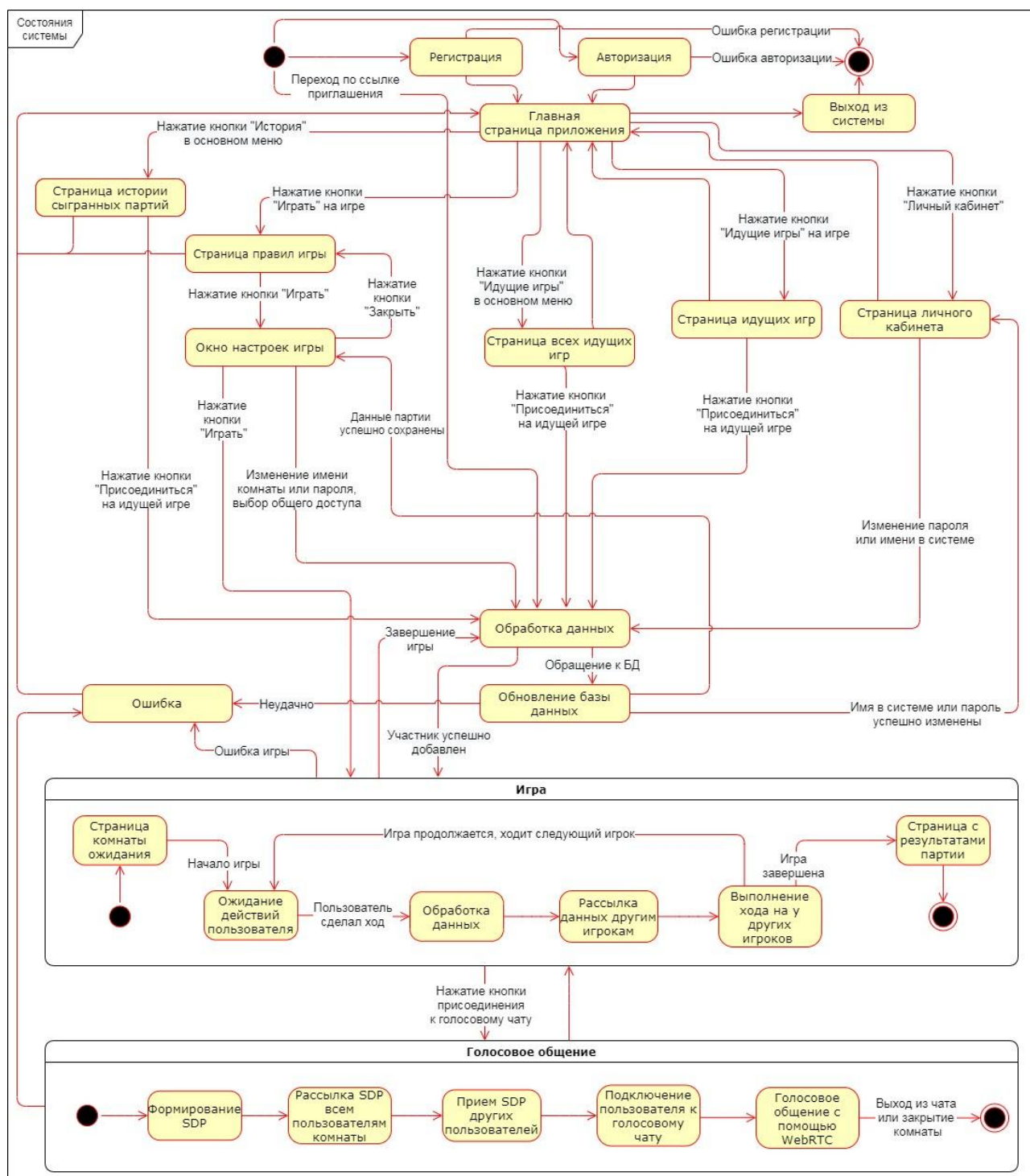


Рис. 22. Диаграмма состояний

2.7.4. Варианты действий в системе

Диаграмма активности, изображенная на рисунке 23, отражает возможные действия, описанные в пункте 3.2.3, состояния которых отображены на диаграмме состояний (рисунок 22).

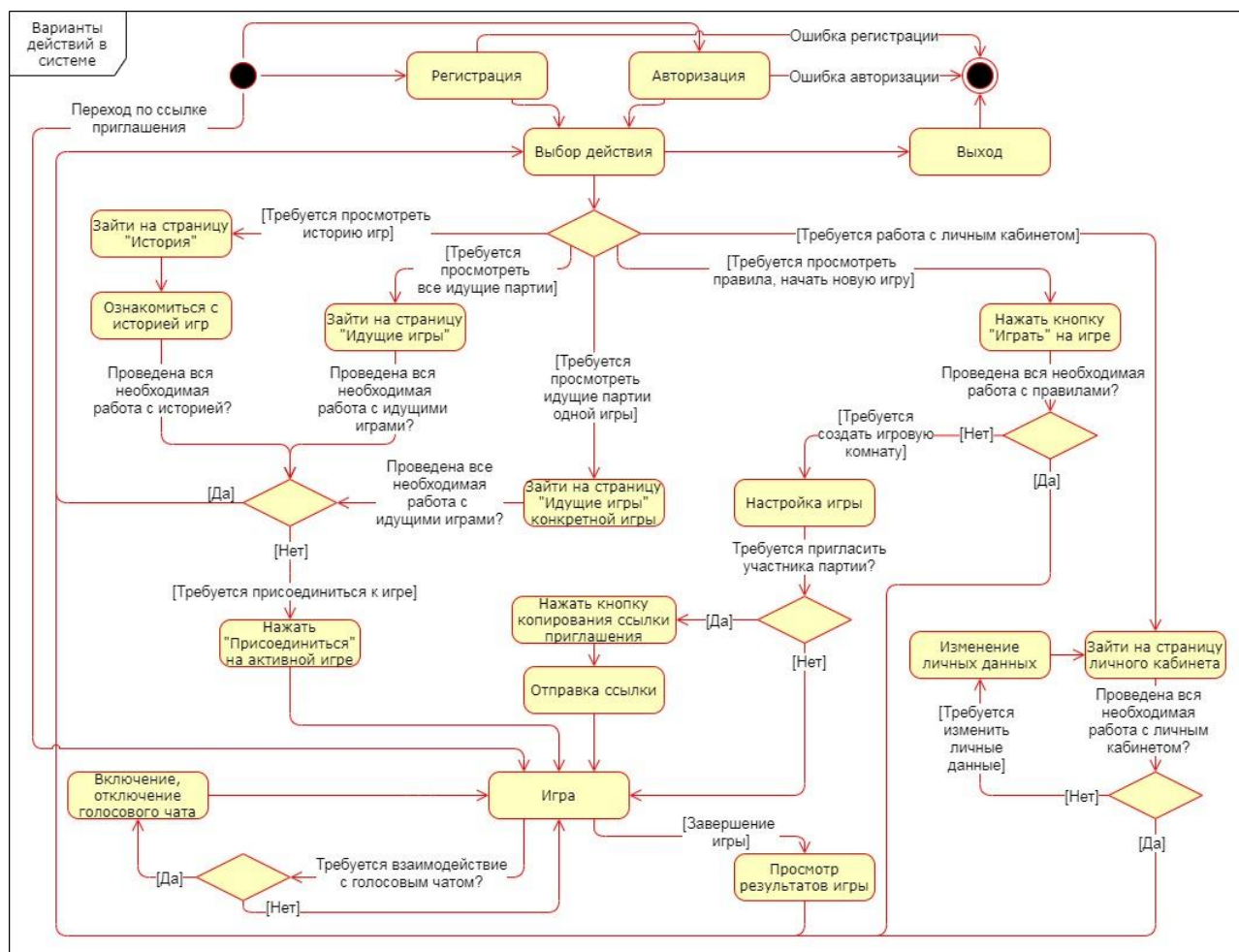


Рис. 23. Диаграмма активности

2.7.5. Развертывание приложения

На рисунке 24 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.

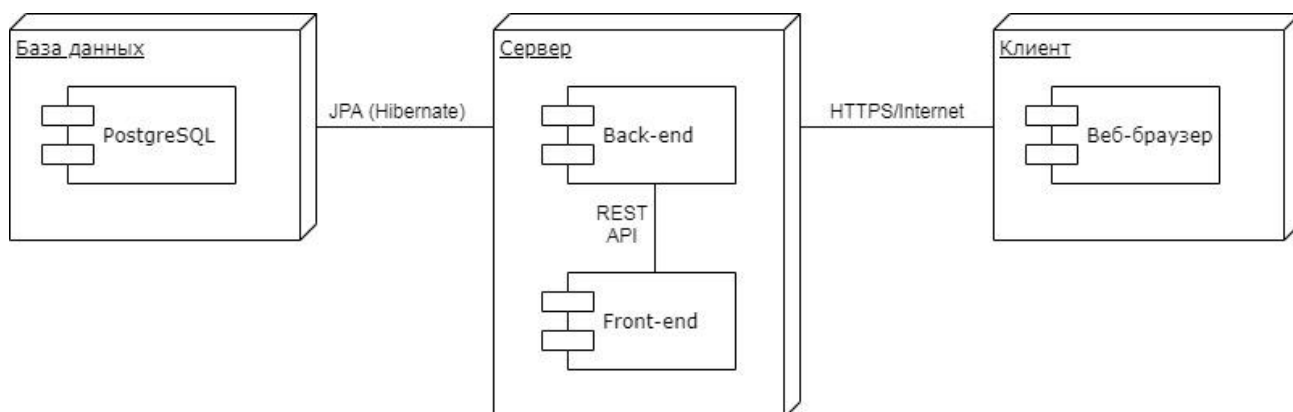


Рис. 24. Диаграмма развертывания

2.7.6. Диаграмма классов

Диаграмма классов приложения изображена на рисунке 25.

В пакете `models` находятся классы, описывающие сущности. Абстрактные классы `GameStep` и `ServerStepResponse` используются для веб-сокетного взаимодействия и изменения состояния игры.

В пакете `models.dto` находятся data transfer object классы, в пакете `models.entities` – классы сущностей для работы с базой данных. В пакете `models.mappers` находятся интерфейсы мапперов для преобразования соответствующих типов из пакетов `models.dto` и `models.entities`.

В пакете `models.repositories` находятся интерфейсы репозиториев, наследуемых от `PagingAndSortingRepository` – интерфейса фреймворка Spring Data, предоставляющего набор методов для работы с БД.

В пакете `services` находятся интерфейсы для работы с сущностями и репозиториями, в `services.impl` находятся классы, имплементирующие соответствующие интерфейсы из пакета `services`.

В пакете `controllers` находятся контроллеры, необходимые для взаимодействия между серверной и клиентской частями приложения через REST API.

В пакете config находятся файлы конфигурации Spring для веб-сокетного взаимодействия и конфигурации CORS.

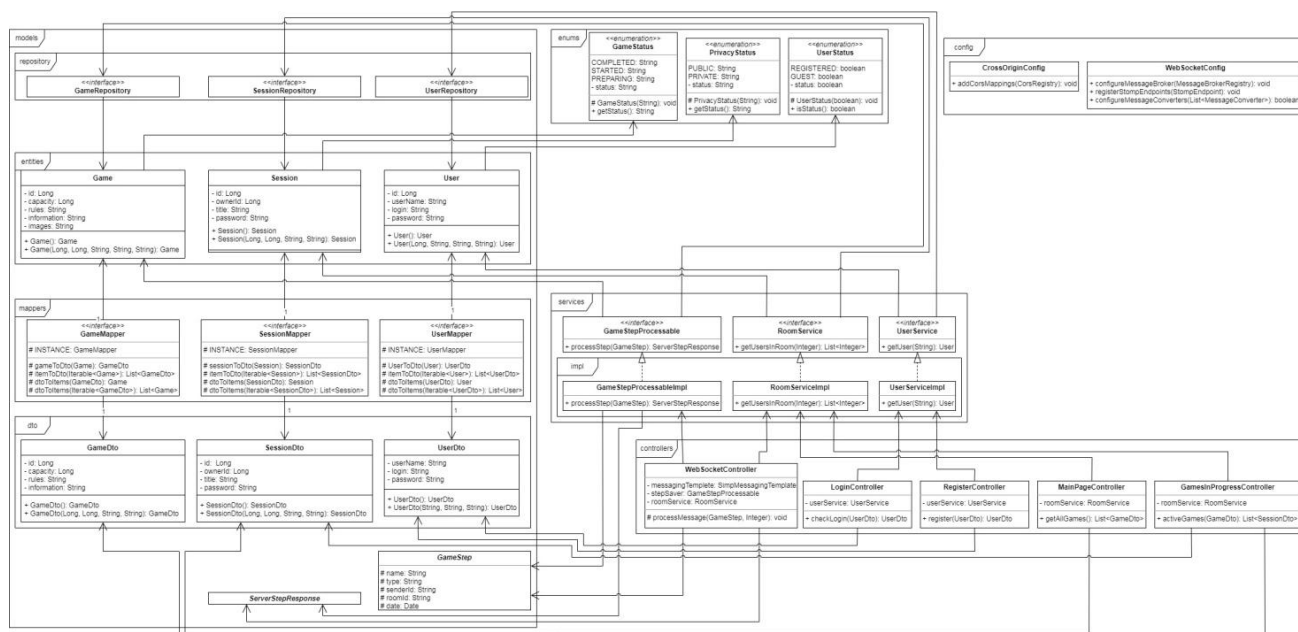


Рис. 25. Диаграмма классов

2.7.7. Диаграмма объектов

Для лучшего понимания диаграммы классов была составлена диаграмма объектов (рисунок 26). На ней отображены объекты классов сущностей в некоторый момент времени.

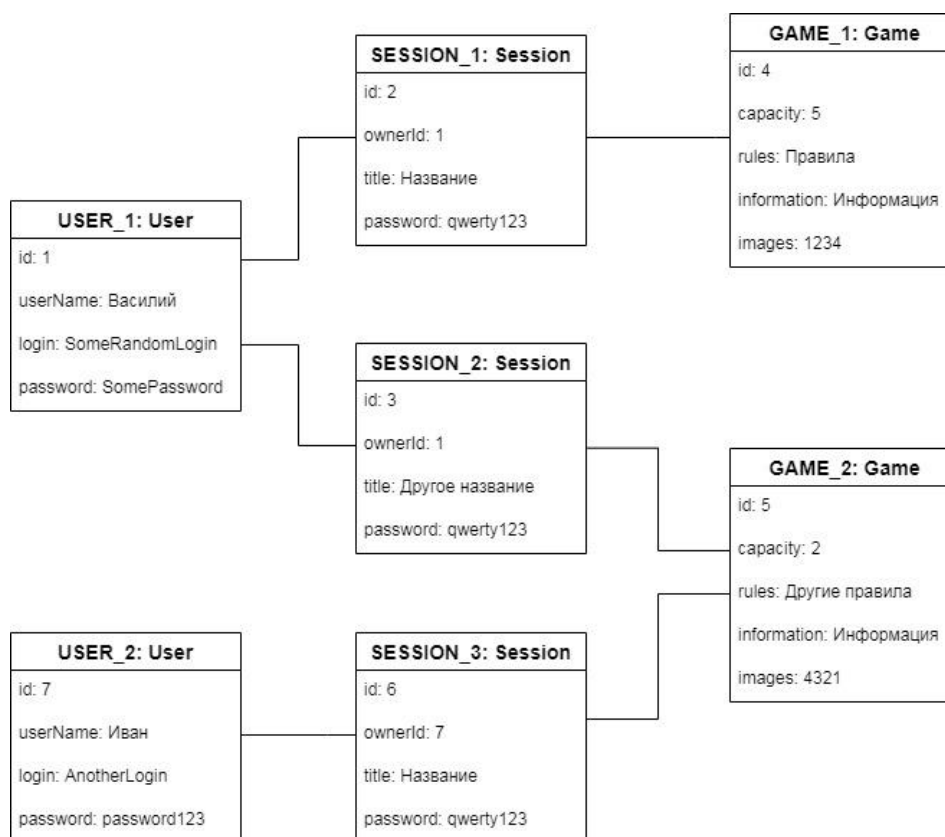


Рис. 26. Диаграмма объектов

2.7.8. IDEF0

На рисунке 27 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 28, узла A4 «Работа с системой» - на рисунке 29.



Рис. 27. IDEF0 диаграмма

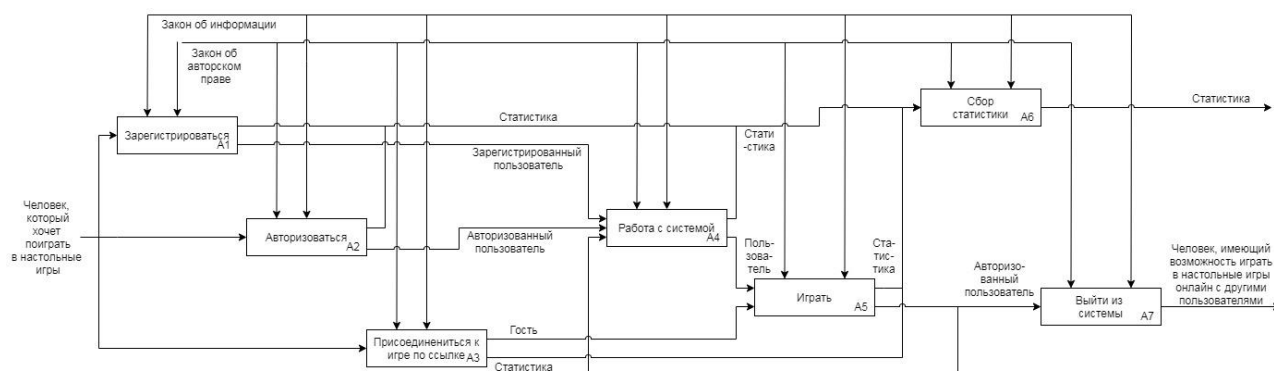


Рис. 28. IDEF0 – расшифровка узла A0 «Работа сервиса»

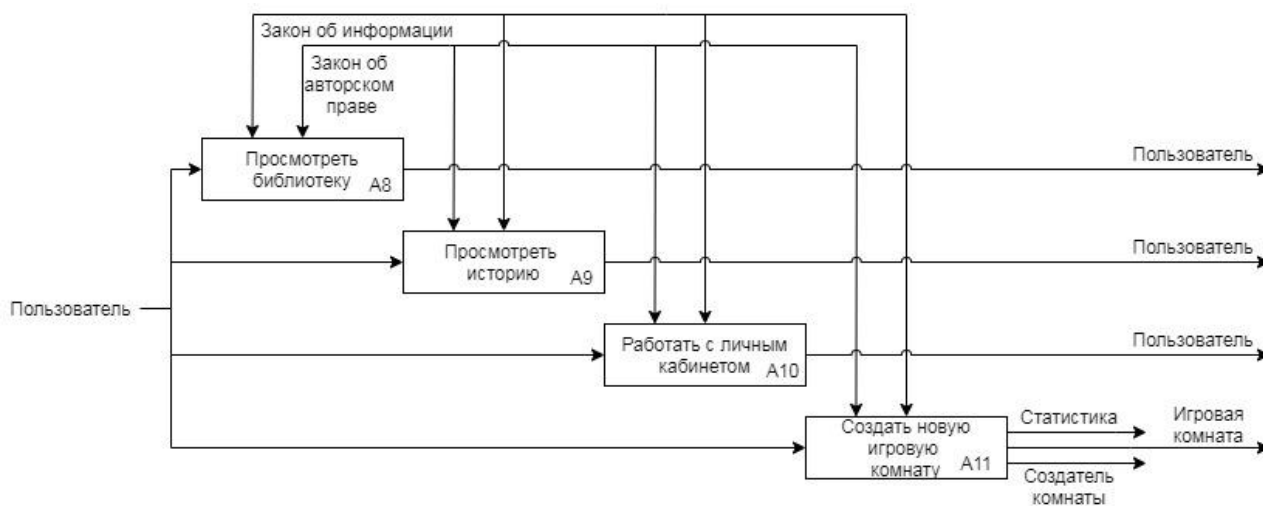


Рис. 29. IDEF0 – расшифровка узла A4 «Работа с системой»

3. Архитектура приложения

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.

4. Проектная часть

4.1 Интерфейс приложения

4.2 Swagger

Swagger - это фреймворк для спецификации Rest-API. Специальный инструмент Swagger UI позволяет создать интерактивную документацию API, которая позволяет проверять вызов API прямо в браузере.

5. Тестирование

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

- smoke testing;
- UI тестирование;
- usability тестирование.

5.1 Smoke testing

5.2 UI тестирование

5.3 Usability тестирование

6. Заключение