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](#_Toc67418154)

[Введение 4](#_Toc67418155)

[1. Постановка задачи 5](#_Toc67418156)

[1.1 Цель 5](#_Toc67418157)

[1.2 Сфера использования 5](#_Toc67418158)

[1.3 Требования 5](#_Toc67418159)

[1.4 Задачи 5](#_Toc67418160)

[2. Анализ предметной области 6](#_Toc67418161)

[2.1 Целевая аудитория 6](#_Toc67418162)

[2.2 Описание предметной области 8](#_Toc67418163)

[2.3 Пользователи системы 8](#_Toc67418164)

[2.4 Границы функциональности 9](#_Toc67418165)

[2.5 Анализ существующих решений 10](#_Toc67418166)

[2.6 Воронки конверсии 15](#_Toc67418167)

[2.7 Анализ задачи 16](#_Toc67418168)

[2.7.1. Варианты использования приложения 16](#_Toc67418169)

[2.7.2. Взаимодействие компонентов системы 16](#_Toc67418170)

[2.7.3. Варианты состояния системы 16](#_Toc67418171)

[2.7.4. Варианты действий в системе 16](#_Toc67418172)

[2.7.5. Развертывание приложения 16](#_Toc67418173)

[2.7.6. Диаграмма классов 16](#_Toc67418174)

[2.7.7. Диаграмма объектов 17](#_Toc67418175)

[2.7.8. ER-диаграмма 17](#_Toc67418176)

[2.7.9. IDEF0 17](#_Toc67418177)

[3. Архитектура приложения 17](#_Toc67418178)

[4. Проектная часть 19](#_Toc67418179)

[4.1 Интерфейс приложения 19](#_Toc67418180)

[4.2 Swagger 19](#_Toc67418181)

[5. Тестирование 19](#_Toc67418182)

[5.1 Smoke testing 19](#_Toc67418183)

[5.2 UI тестирование 19](#_Toc67418184)

[5.3 Usability тестирование 19](#_Toc67418185)

[6. Заключение 20](#_Toc67418186)

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.

# Постановка задачи

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

## Цель

Разработать веб-приложение настольных игр, которое:

* Позволит как зарегистрированным, так и незарегистрированным пользователям принимать участие в играх;
* Позволит участникам игры общаться с помощью голосовой связи;
* Позволит пользователю просматривать правила игры и статистику по сыгранным партиям.

## Сфера использования

Повседневная жизнь.

## Требования

Разрабатываемое веб-приложение должно предоставлять пользователям следующие возможности:

1. Регистрация и авторизация пользователей;
2. Возможность редактирование профиля пользователя;
3. Возможность пользователя принимать участие в существующих играх, приглашать неавторизованных пользователей для принятия участия в игре в качестве активного игрока или наблюдателя;
4. Возможность общения игроков с помощью голосовой связи;
5. Возможность пользователя просматривать правила игры и статистику по сыгранным партиям.

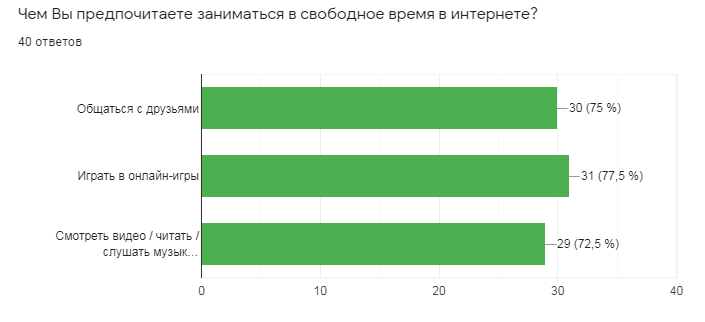
## Задачи

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.

# Анализ предметной области

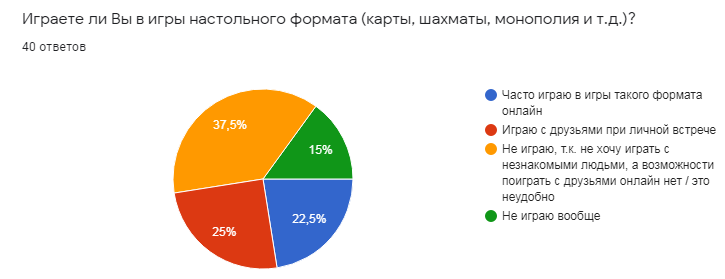
## Целевая аудитория

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.).



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.).



1. 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.).



1. 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 интернет.

## Описание предметной области

Написать инфу зачем сервис, что делает, в общих чертах

## 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) 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;
* Переговариваться с другими участниками партии с помощью голосовой связи;
* Просмотр результатов партии;
* Просмотр общей статистики (количество проведенных игр, количество побед и т.п.).

## 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.

## 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.

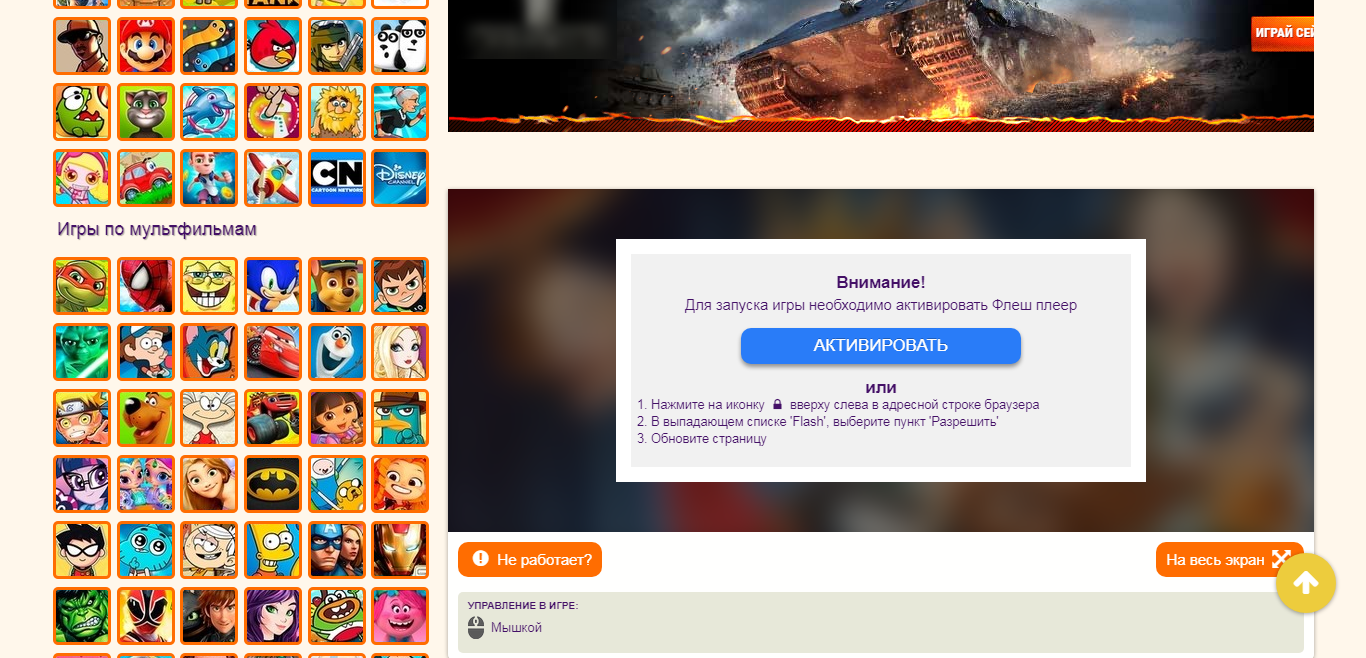
Чтобы сделать анализ аналогов более конкретным, рассмотрим реализацию игры «Каркассон», как первого продукта на нашей платформе.

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.

* Сервис «ИГРОУТКА»

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



1. Интерфейс сервиса «ИГРОУТКА»

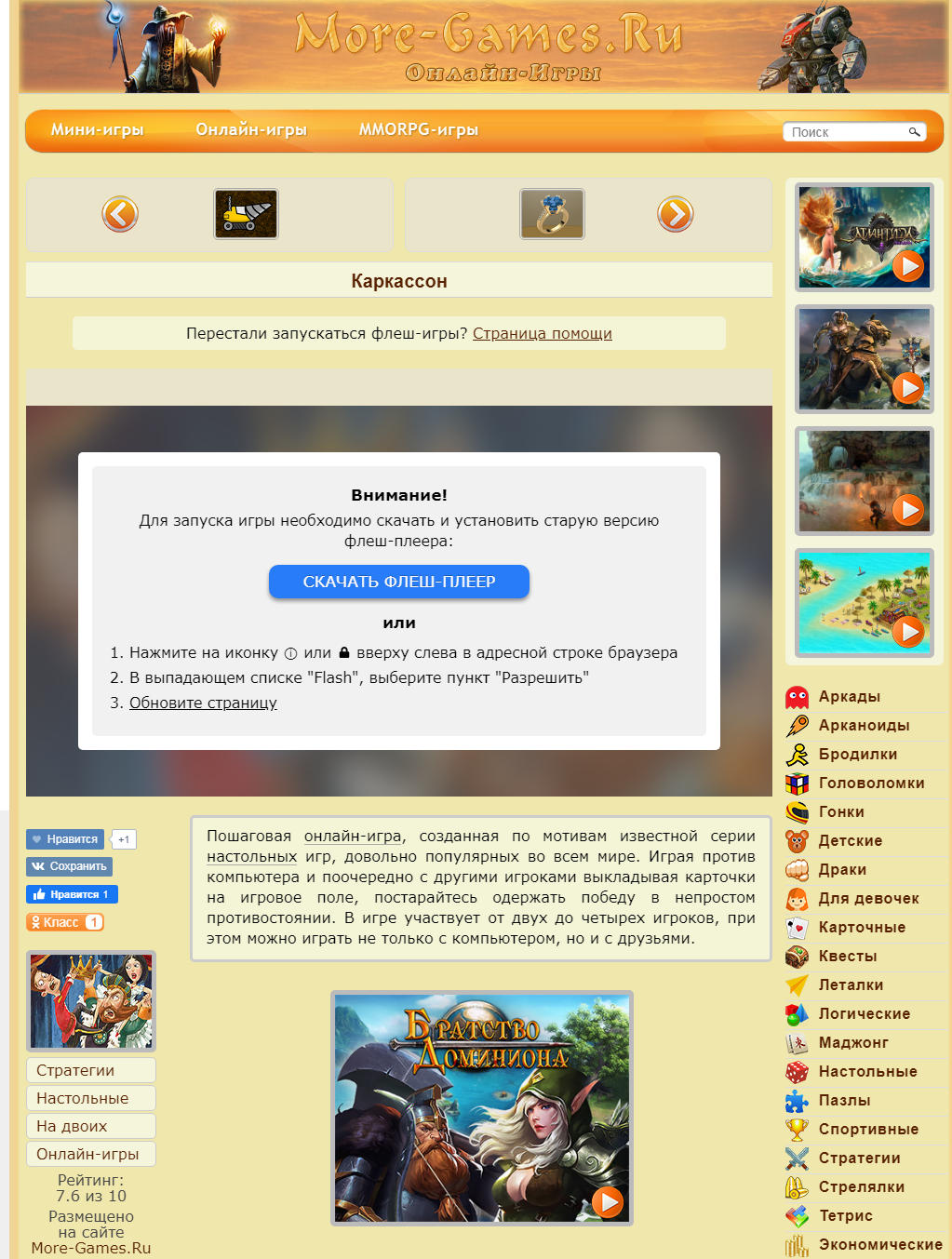
Достоинства:

* 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.
* Сервис «More-Games.Ru»

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



1. Интерфейс сервиса «More-Games.Ru»

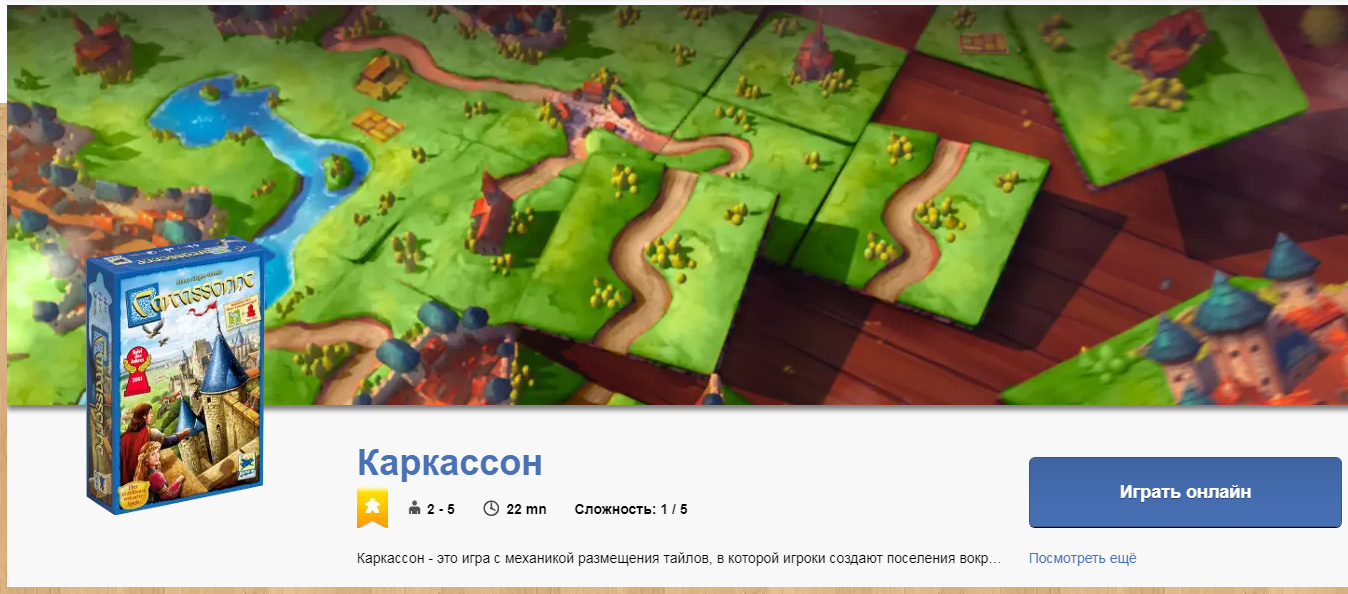
Достоинства:

* 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.
* Сервис «BoardGameArena»

[https://boardgamearena.com/gamepanel?game=carcassonne](https://vk.com/away.php?to=https%3A%2F%2Fboardgamearena.com%2Fgamepanel%3Fgame%3Dcarcassonne&cc_key=)



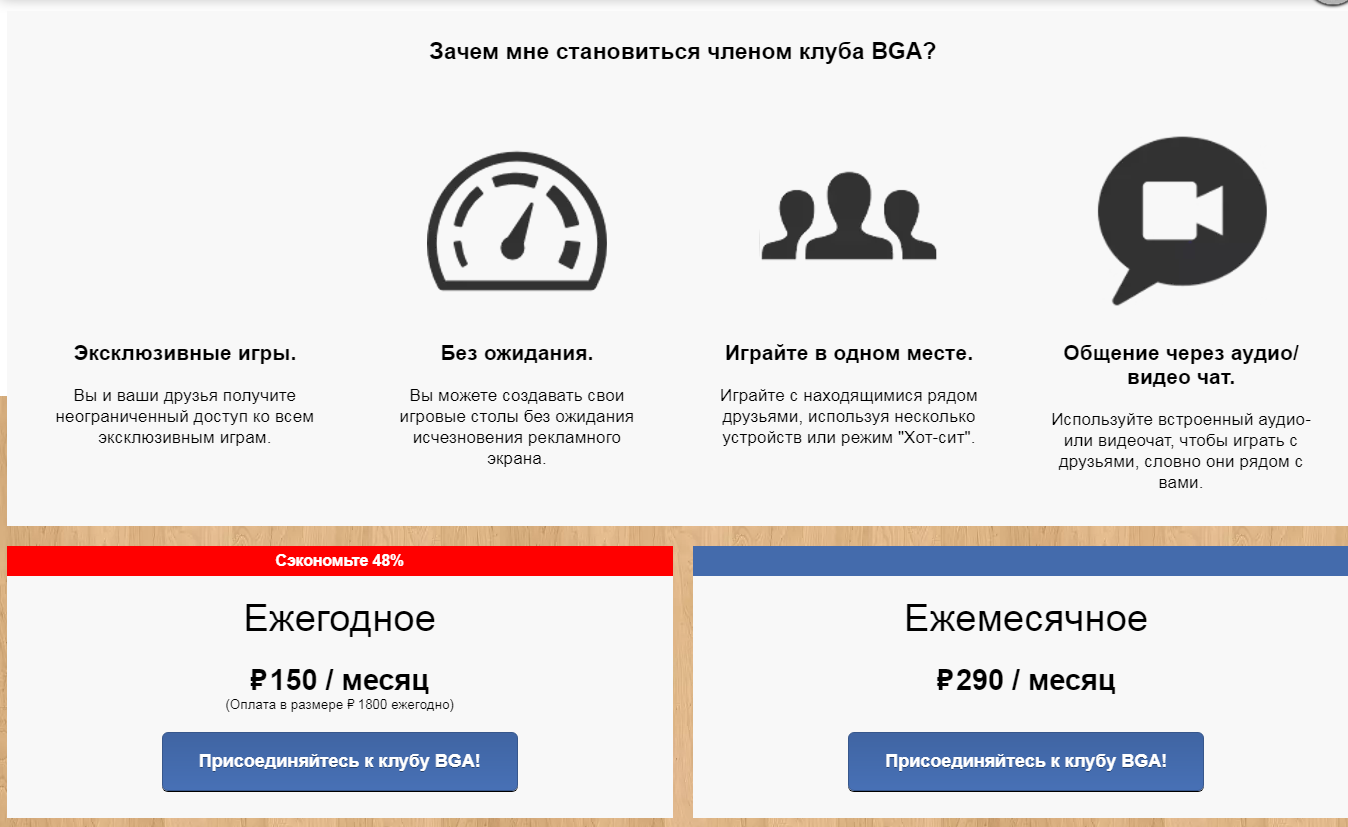
1. Интерфейс сервиса «BoardGameArena»

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 аудиочат.

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.



1. 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

## 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.

Здесь будет информация по воронкам конверсии для разных действий

## Анализ задачи

# Варианты использования приложения

Use-Case диаграмма и описание действий каждой группы пользователей

# Взаимодействие компонентов системы

Диаграммы последовательностей и диаграммы взаимодействий для разных действий

# Варианты состояния системы

Диаграмма состояния

# Варианты действий в системе

Диаграмма активности

# Развертывание приложения

Диаграмма развертывания

# Диаграмма классов

Диаграмма классов

# Диаграмма объектов

Диаграмма объектов

# ER-диаграмма

ER-диаграмма

# IDEF0

EDEF0

# 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, в особенности таких, где требуется интенсивно работать с формами.

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 нам 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 JavaScript, 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.

Специфика клиентской стороны веб-приложения требует отладки кода во всех популярных браузерах. Наше приложение разрабатывалось под браузер Google Chrome. Этот браузер, основан на WebKit, обладает встроенным инструментом разработки Web Inspector, который очень хорошо развит и позволяет выполнять отладку JavaScript кода.

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

Сюда дописывать про используемые технологии

# Проектная часть

## Интерфейс приложения

Скриншотики страниц с описанием

## 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.

Информация по нашему сваггеру

# Тестирование

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

* Smoke testing;
* UI тестирование;
* Usability тестирование.

Или другие

Инфа по тестированиям

## Smoke testing

## UI тестирование

## Usability тестирование

# Заключение

Надо написать