Оглавление

[1. Концепция 2](#_Toc62485839)

[1.1 Описание 2](#_Toc62485840)

[2. Пользовательский интерфейс 2](#_Toc62485841)

[2.1 Главное меню . 2](#_Toc62485842)

[2.2 Одиночная игра . 2](#_Toc62485843)

[2.3 Многопользовательская игра . 4](#_Toc62485844)

[2.4 Режимы игры (модификации) . 4](#_Toc62485845)

[2.4 Статистика 4](#_Toc62485846)

[2.5 Настройки 5](#_Toc62485847)

[3. Клиент 5](#_Toc62485848)

[3.1 FrontEnd 5](#_Toc62485849)

[3.2 BackEnd 5](#_Toc62485850)

[4. Сервер 5](#_Toc62485851)

[5. База данных . 5](#_Toc62485852)

# Концепция.

## Описание.

Основной идеей является реализация головоломки с возможностью настройки режимов игры и возможностью играть по сети.

Многопользовательская игра по сети предполагается как совместная, так и друг против друга на время. Время будет вынесено отдельным параметром в настройки, чтобы те, кому оно не нужно могли на него не отвлекаться. Также предполагается создание рейтингового режима с рейтингом ЭЛО для самых популярных режимов игры. Делать это для всех возможных настроек не целесообразно. Суть рейтинговой игры будет заключаться в том, чтобы быстрее своего противника собрать одинаковую головоломку.

По сути, основным плюсом будет то, что в одной игре будет собрано несколько подобных головоломок и наличие сетевой игры, которая часто интереснее обычной.

# Пользовательский интерфейс.

## 2.1 Главное меню.

Главное меню приложения, должно содержать кнопки.

1. Один игрок.
2. Многопользовательская игра.
3. Статистика.
4. Настройки.

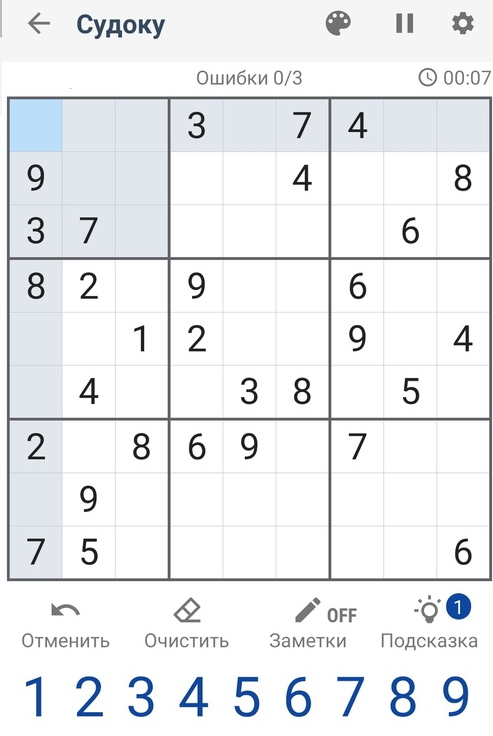
## Одиночная игра.

Игра в одиночку в соответствии с выбранным в настройке режимом. Пользователь должен иметь возможность играть без подключения к серверу и иметь такую же функциональность, как при игре с сервером.

В процессе игры пользователь видит:

* Игровое поле.
* Различные особенности игрового поля связанные с выбранной модификацией (например, суммы ячеек).
* Время сбора головоломки.
* Режим ввода (пометки или ответы).
* Числа.
* Выделения зон с уникальными числами (строки, столбцы, блоки).
* Количество сделанных ошибок (настраиваемый параметр).
* Ошибся ли он при вводе (настраиваемый параметр).

Ниже, изображение из существующего приложения. Основные идеи предлагается оставить такими же, но переработать дизайн и нужно подумать, что можно улучшить.



Для браузерной версии серьёзным улучшением может стать использование клавиатуры и использование горячих клавиш для установки ответа/пометки.

В процессе игры пользователь может:

* Вводить ответы в ячейку.
* Вводить пометки в ячейку.
* Отменять действия.
* Сдаться и посмотреть сетку.
* Очистить ячейку от заметок или числа.

## Многопользовательская игра.

В многопользовательской игре сохраняется всё, что есть в однопользовательской игре.

Также появляется возможность общения (чат, возможно звуковой чат).

В перспективе возможно создание работы приложения в режиме сервера для работы в локальной сети или через Blutooth.

## 2.4 Режимы игры (модификации).

В игре необходимо окно настройки текущего режима игры. Настраиваемыми параметрами являются:

* Тип игры. Судоку – класика. Сумдоку – суммы и так далее.
* Размер игрового поля. (9,16).
* Учёт диагоналей.
* Наличие блоков с уникальными числами.
* Форма блоков с уникальными числами (квадрат, или произвольная фигура).

Отдельно стоит выделить настройку сложности.

Имея такие настройки, можно будет делать интересные режимы для сборки, но прежде всего, необходимо реализовать классические варианты, а потом уже дополнять их чем-то новым.

## Статистика.

В статистике пользователю можно выводить следующие параметры:

* Общее наигранное время.
* Лучшее время для рейтинговых режимов.
* Любимый режим.
* Рейтинг.
* Максимальное количество побед подряд.
* Текущее количество побед подряд.
* Максимальное количество поражений подряд.
* Текущее количество поражений подряд.

Также можно вести статистику по игре с друзьями. Статистику личных встреч с конкретными игроками.

## Настройки.

* Настройки стиля.
* Показывать сделанные ошибки.
* Показывать количество сделанных ошибок.
* Наличие игрового времени.
* Наличие подсказок.

## Перечисление функций пользовательского интерфейса.

Игровые:

* Новая игра.
* Сдаться.
* Ввести число.
* Ввести пометку.
* Получить игровое поле.
* Получить игровое время.
* Отменить действие.
* Стереть пометки.
* Получить подсказку.
* Получить количество ошибок.
* Выход из игры с сохранением всех её параметров.

Общие:

* Получить версию игры.
* Получить информацию о сайте проекта.
* Получить описание проекта.
* Получить обучение (Нужно расширить.)

Настройки:

* Получить пользовательские настройки (не вижу смысла передавать или получать настройки по отдельности, лучше одной структурой).
* Изменить пользовательские настройки (не вижу смысла передавать или получать настройки по отдельности, лучше одной структурой).
* Получить описание отдельной настройки.

Взаимодействие с сервером:

* Регистрация
* Вход в учётную запись.
* Смена пароля.
* Восстановление пароля.
* Создание лобби.
* Вход в лобби.
* Выход из лобби.
* Получение списка лобби.
* Получение описания лобби.
* Получение списка игроков.
* Получение статистики и достижений любого игрока.
* Получение общей статистики, достижений и рейтинга.
* Поиск противников для игры.
* Обновить игру (с описанием проекта, информацией о сайте и так далее).
* Получение статистики и достижений.
* Получение рейтинга.
* Функции чата для игроков.
* Чат в лобби и сеансе игры.

# Клиент.

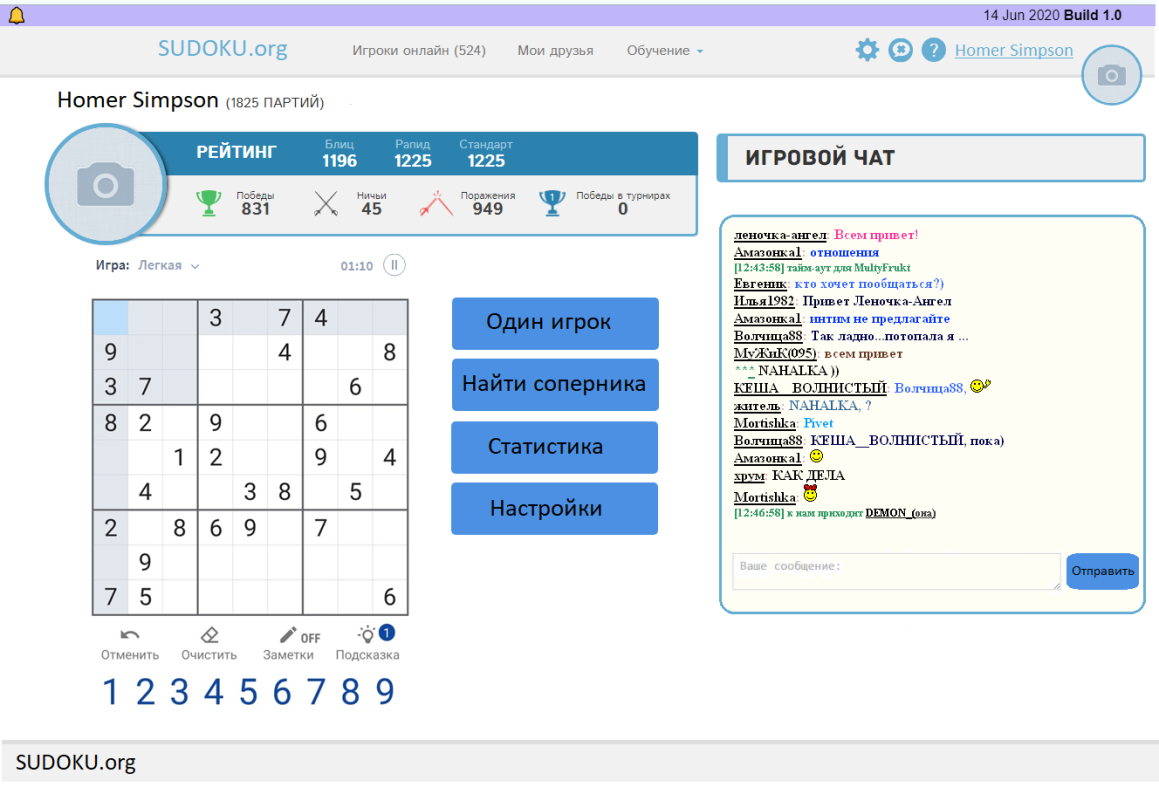
## 3.1 UI requirements.

Для разработки пользовательского интерфейса приложения используется сборщик модулей webpack v4.42.1. Компонентный подход реализован на фреймворке AngularJS v1.8.2.

## 3.1.1 Пользовательский интерфейс для браузерной версии приложения.

Основным подходом при описании данного раздела будет описание конкретной страницы приложения с подробным описанием элементов страницы и функционалом. Также возможны некоторые вложения типа картинок, диаграмм, таблиц, аппликаций с примерами визуального оформления отдельных элементов, которые являются лишь макетом для передачи наиболее полного понимания той или иной части приложения и не обязаны точно соответствовать действительной реализации в приложении.

## 3.1.1.1 Главная страница.



## 3.1.1.2 Настройки.

## 3.1.1.3 Статистика.

## 3.1.2 Пользовательский интерфейс для мобильной версии приложения.

## 3.2 BackEnd.

## 3.3 Роутинг и маршрутизация.

В данном разделе необходимо описать основные механизма реализации перехода между различными частями приложения, а также описать структуру в виде списка с наименованиями отдельных частей/страниц.

# Сервер.

# База данных.