"Многопользовательская игровая платформа"

Тема - шашки

Группа: 09-422

Аналитик: Гараева Диляра

Разработчик: Шайхутдинова Алина

Тестировщик: Новоселова Ульяна

Оглавление

1. [ОБЗОР СИСТЕМЫ 3](#_Toc199954334)

[1.Функциональные требования 3](#_Toc199954335)

[2. Требования к логической структуре БД 21](#_Toc199954336)

# 1. ОбЗОР Системы

1.1 Функциональные требования

1**.User Story**: Авторизация

Как зарегистрированный пользователь, я хочу войти в свой аккаунт, чтоб получить доступ ко всему функционалу приложения.

1.2 **Пользователь:**

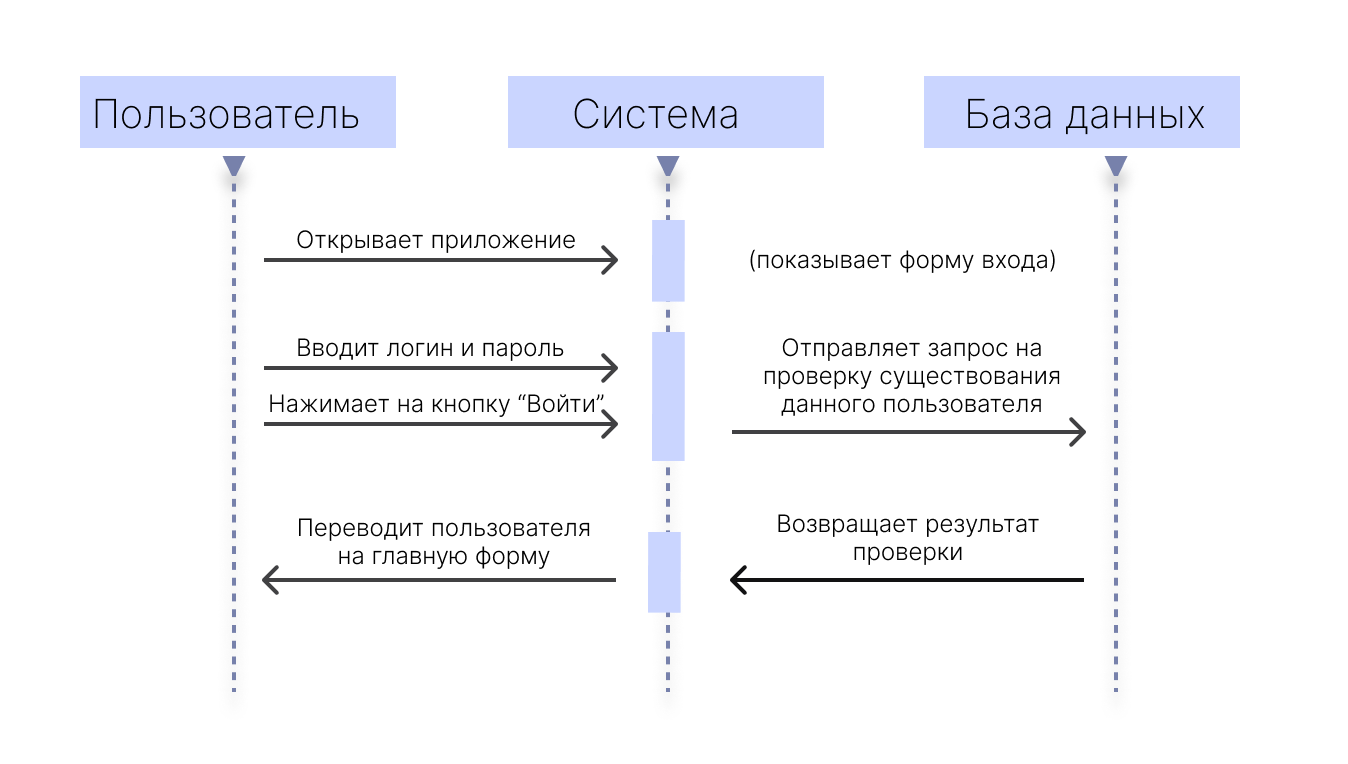
* **Зарегистрированный пользователь** может авторизоваться и получать доступ к главной форме приложения
* **Незарегистрированный пользователь (Гость)** может перейти на форму регистрации

1.3 **Место:**

* Форма входа
* Форма регистрации

1.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать дизайн формы входа |
| 2 | Добавить валидацию полей формы |
| 3 | Настроить отправку данных в систему при нажатии на кнопку «Войти» |
| 4 | Обработать ответ от сервиса |
| 5 | Связать форму входа с формой регистрации |

1.5 **Диаграмма последовательностей**

1.6 **Техническое задание**

1. Реализовать дизайн формы входа

Основные элементы:

1. Подзаголовок

Тип элемента: текст

1. Поля для ввода данных пользователем (логин, пароль)

Тип элемента: текст

1. Кнопка «Войти»

Функция: переход на главную страницу

Тип элемента: кнопка

Цвет: зеленый

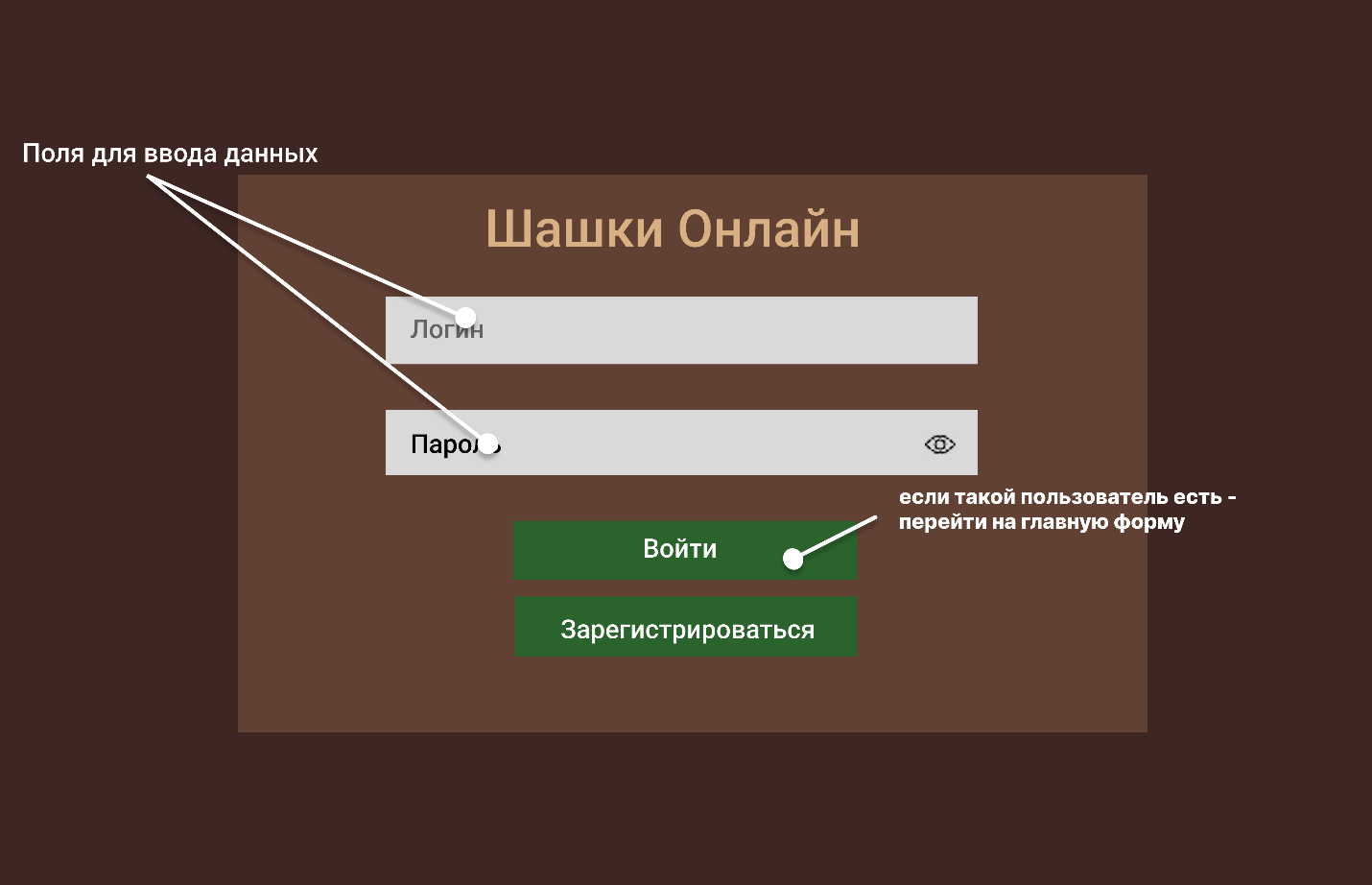
1. Кнопка «Зарегистрироваться»

Функция: переход на страницу регистрации

Тип элемента: кнопка

Цвет: зеленый

Макет формы входа



1. Добавить валидацию полей формы
2. Добавить проверку на заполненность всех полей
3. Настроить отправку данных в систему при нажатии на кнопку «Войти»

При нажатии на кнопку «Войти»:

1. Проверить корректность введенных данных(логина и пароля)
2. Хэширование пароля
3. Отправить запрос на аутентификацию в сервер
4. Обработать ответ от сервиса
5. При успешной аутентификации перевести пользователя на главную форму
6. Если аутентификация прошла неуспешно, вывести ошибку
7. Связать форму входа с формой регистрации
8. При нажатии на кнопку «Зарегистрироваться» перевести пользователя на форму регистрации

2**.User Story**: Регистрация

Как незарегистрированный пользователь, я хочу иметь возможность зарегистрироваться, чтоб получить доступ ко всему функционалу приложения.

2.2 **Пользователь:**

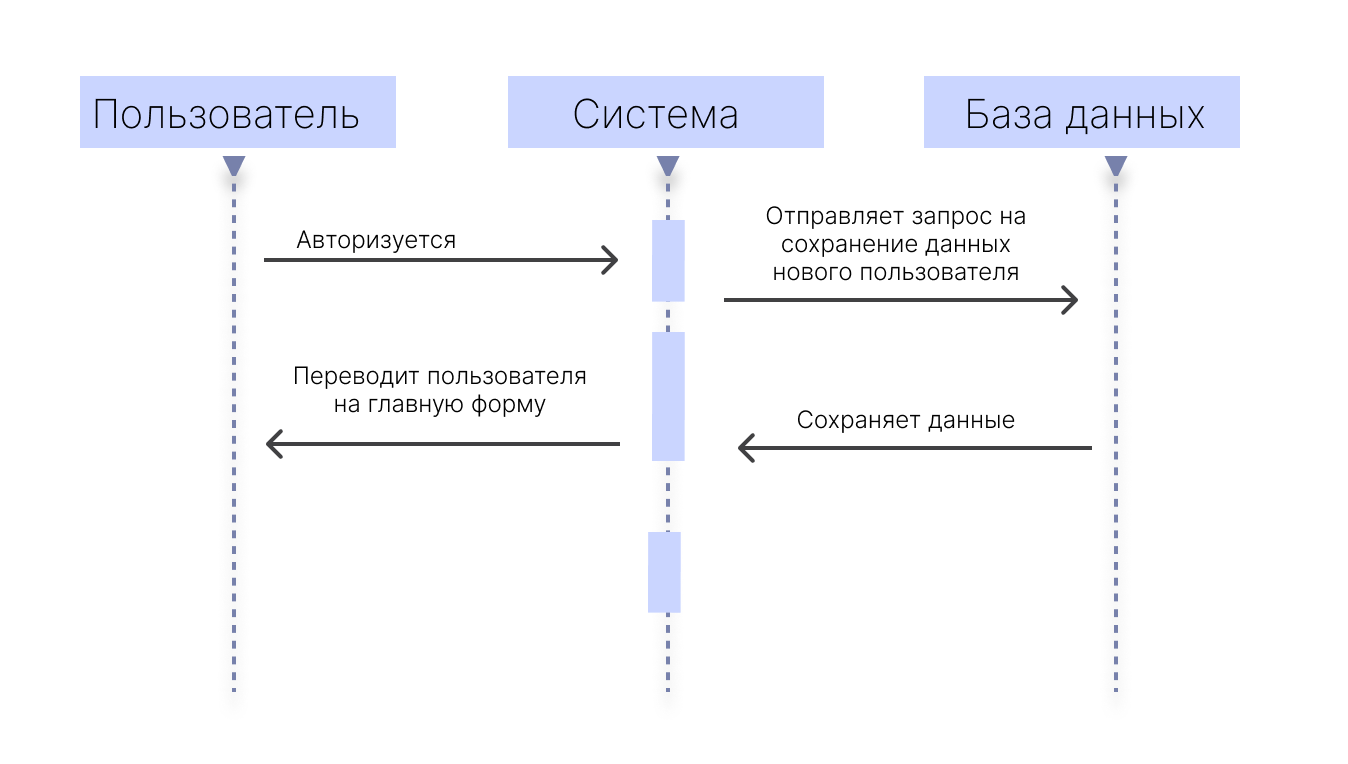
* **Незарегистрированный пользователь (Гость)** может пройти регистрацию

2.3 **Место:**

* Форма регистрации

2.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать дизайн формы входа |
| 2 | Добавить валидацию полей формы |
| 3 | Настроить отправку данных в систему при нажатии на кнопку «Зарегистрироваться» |
| 4 | Обработать ответ от сервиса |
| 5 | Связать форму регистрации с главной формой |
| 6 | Реализовать переход на форму входа при нажатии на кнопку «Назад» |

* 1. **Диаграмма последовательностей**

2.6 **Техническое задание**

1. Реализовать дизайн формы входа

На форму нужно добавить поля для ввода данных:

Основные элементы:

1. Подзаголовок

Тип элемента: текст

1. Поля для ввода данных пользователем (логин, пароль, подтверждение пароля)

Тип элемента: текст

1. Кнопка «Зарегистрироваться»

Функция: переход на главную страницу

Тип элемента: кнопка

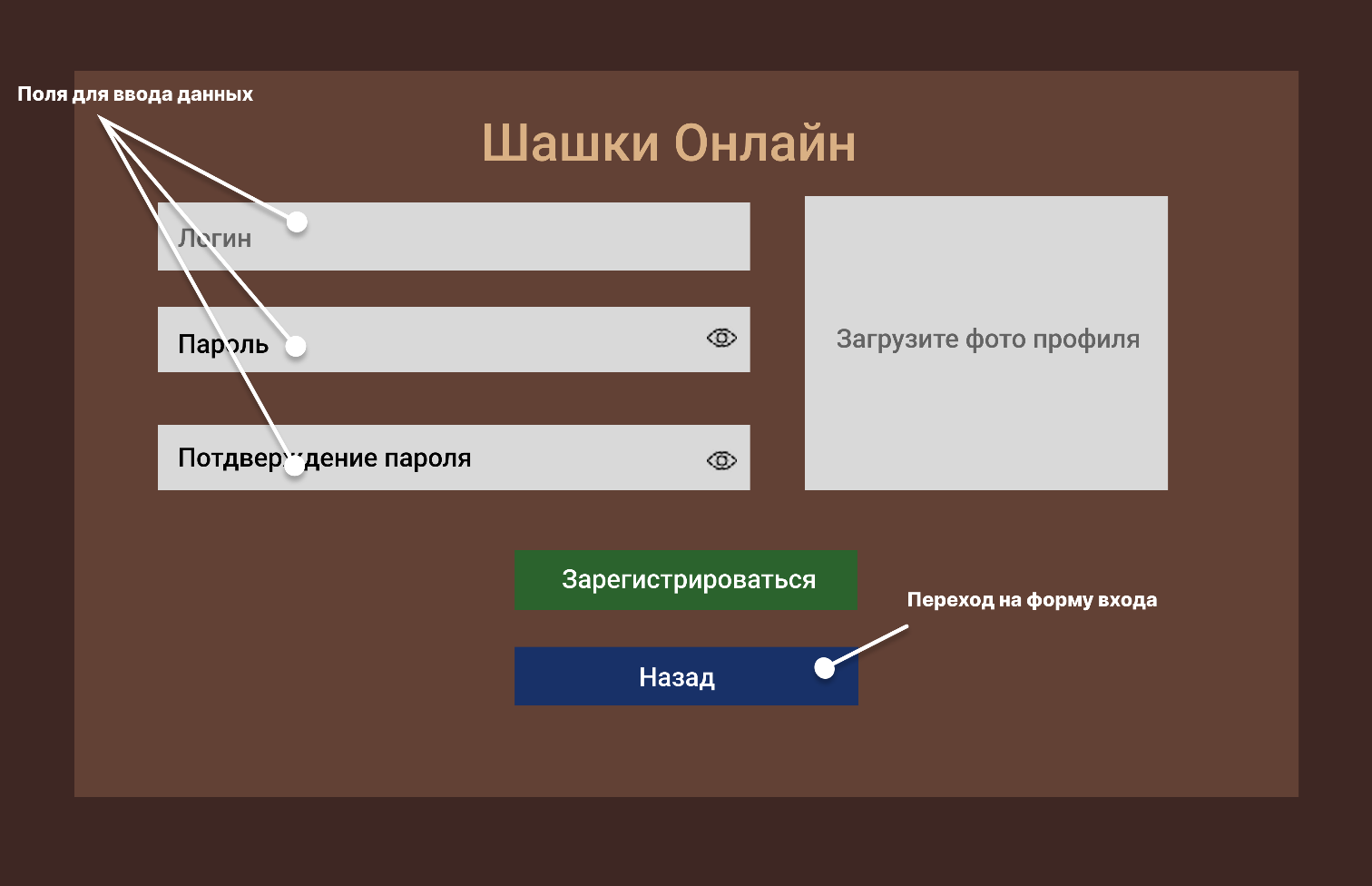
Цвет: зеленый

1. Кнопка «Назад»

Функция: переход на форму входа

Тип элемента: кнопка

Цвет: синий

Макет формы регистрации

1. Добавить валидацию полей формы
2. Добавить проверку на заполненность всех полей
3. Логин должен быть уникальным
4. Пароли должны совпадать
5. Отображение ошибок под соответствующими полями
6. Настроить отправку данных в систему при нажатии на кнопку «Зарегистрироваться»

При нажатии на кнопку «Зарегистрироваться»:

1. Проверить корректность введенных данных(логина и пароля)
2. Отправка данных пользователя в систему
3. Хэширование пароля
4. Сохранение данных в БД
5. Обработать ответ от сервиса
6. При успешной регистрации перевести пользователя на главную форму
7. Если регистрация прошла неуспешно, вывести сообщение об ошибке
8. Связать форму регистрации с главной формой
9. При нажатии на кнопку «Зарегистрироваться» перевести пользователя на главную форму
10. Реализовать переход на форму входа при нажатии на кнопку «Назад»
11. При нажатии на кнопку «Назад» осуществить переход на форму входа

3**.User Story**: Профиль

Как зарегистрированный пользователь, я хочу иметь возможность просматривать свой профиль, чтобы получить доступ к моей статистике.

3.2 **Пользователь:**

* **Зарегистрированный пользователь** может просмотреть свой профиль

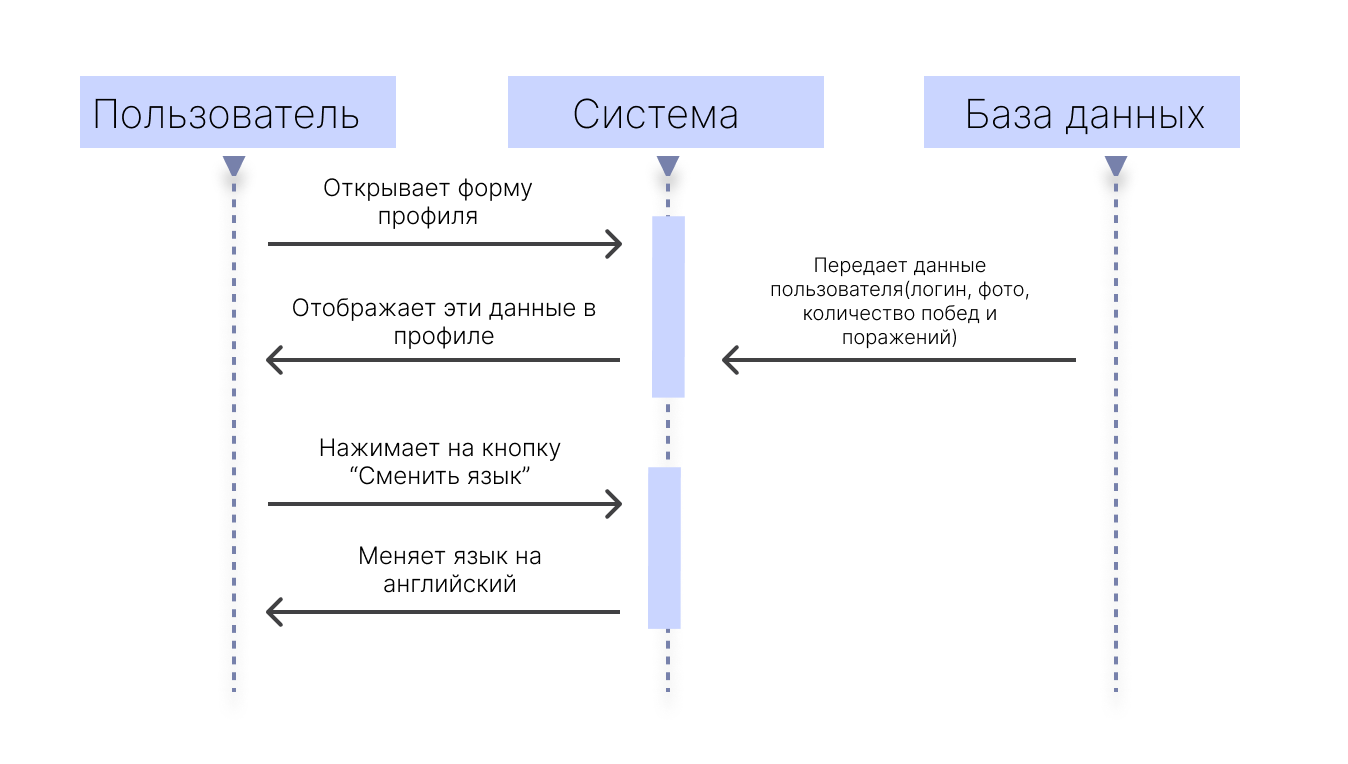
3.3 **Место:**

* Форма профиля

3.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать дизайн формы входа |
| 2 | Отобразить данные пользователя |
| 3 | Реализовать функционал смены языка на английский |
| 4 | Связать форму профиля с главной формой |

* 1. **Диаграмма последовательностей**

****

3.6 **Техническое задание**

1. Реализовать дизайн формы

Основные элементы:

1. Заголовок “Профиль”

Тип элемента: текст

1. Иконки пользователей

Функция: показать фото профилей

Тип элемента: фото

1. Статистика пользователя

Функция: показать количество побед и проигрышей, игр.

Тип элемента: панель

Цвет: бежевый

1. Кнопка «Сменить язык»

Функция: меняет язык приложения на английской

Тип элемента: кнопка

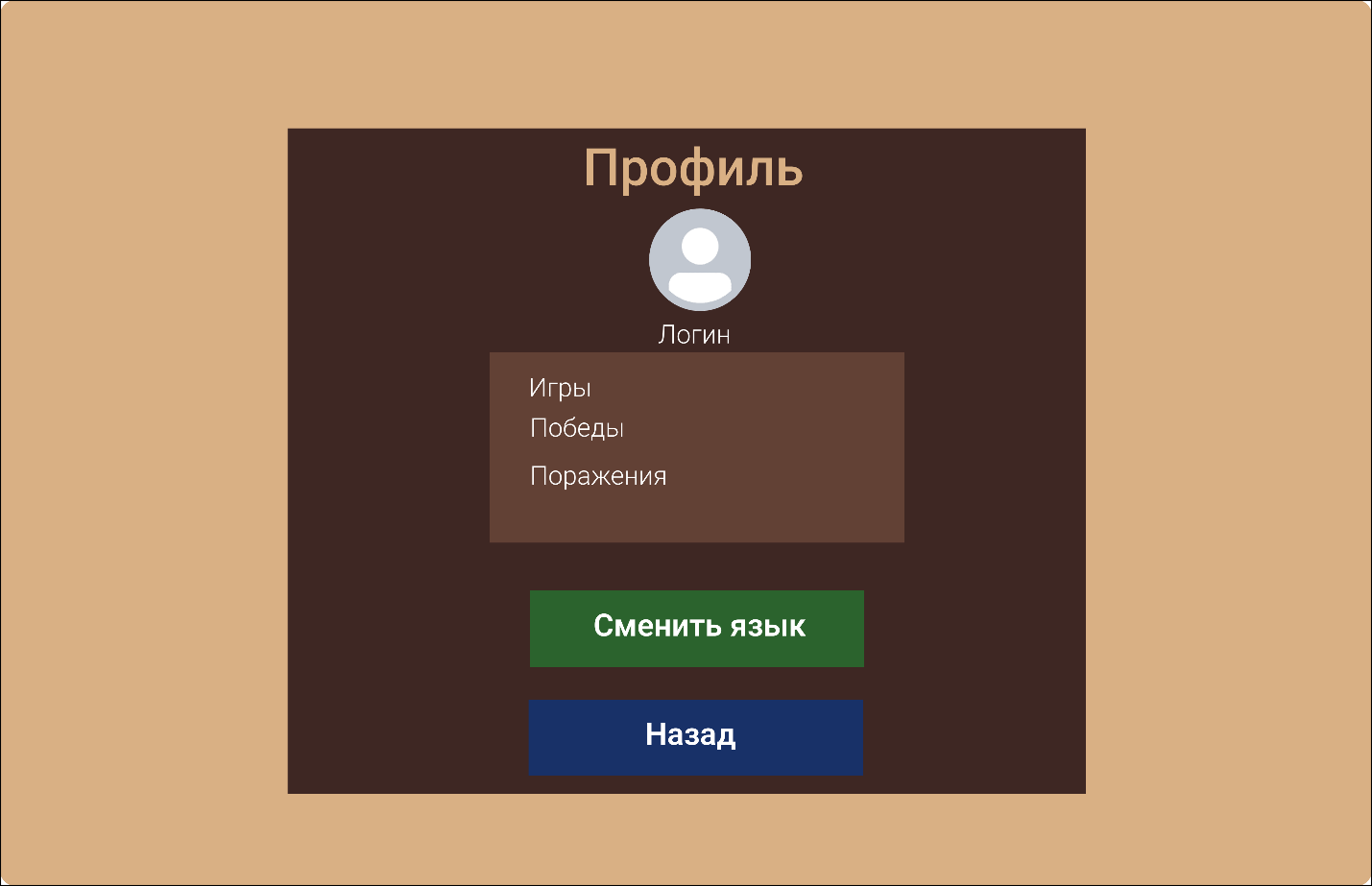
Цвет: зеленый

1. Кнопка «Назад»

Функция: перейти на главную форму

Тип элемента: кнопка

Цвет: синий

Макет формы профиля

1. Отобразить данные пользователя
2. Получить данные пользователя (логин, игры, победы, поражения)
3. Отобразить в профиле
4. Реализовать функционал смены языка на английский

При нажатии на кнопку «Сменить язык»:

1. Изменить язык интерфейса на английский
2. Связать форму профиля с главной формой
3. При нажатии на кнопку «Назад» реализовать переход на главную форму приложения

4**.User Story**: Главная страница

Как зарегистрированный пользователь, я хочу иметь возможность видеть главное меню после входа в систему, чтобы получить доступ к основному функционалу приложения

4.2 **Пользователь:**

* **Зарегистрированный пользователь** может просмотреть свой профиль и рейтинг, создать новую игру, присоединиться к игре

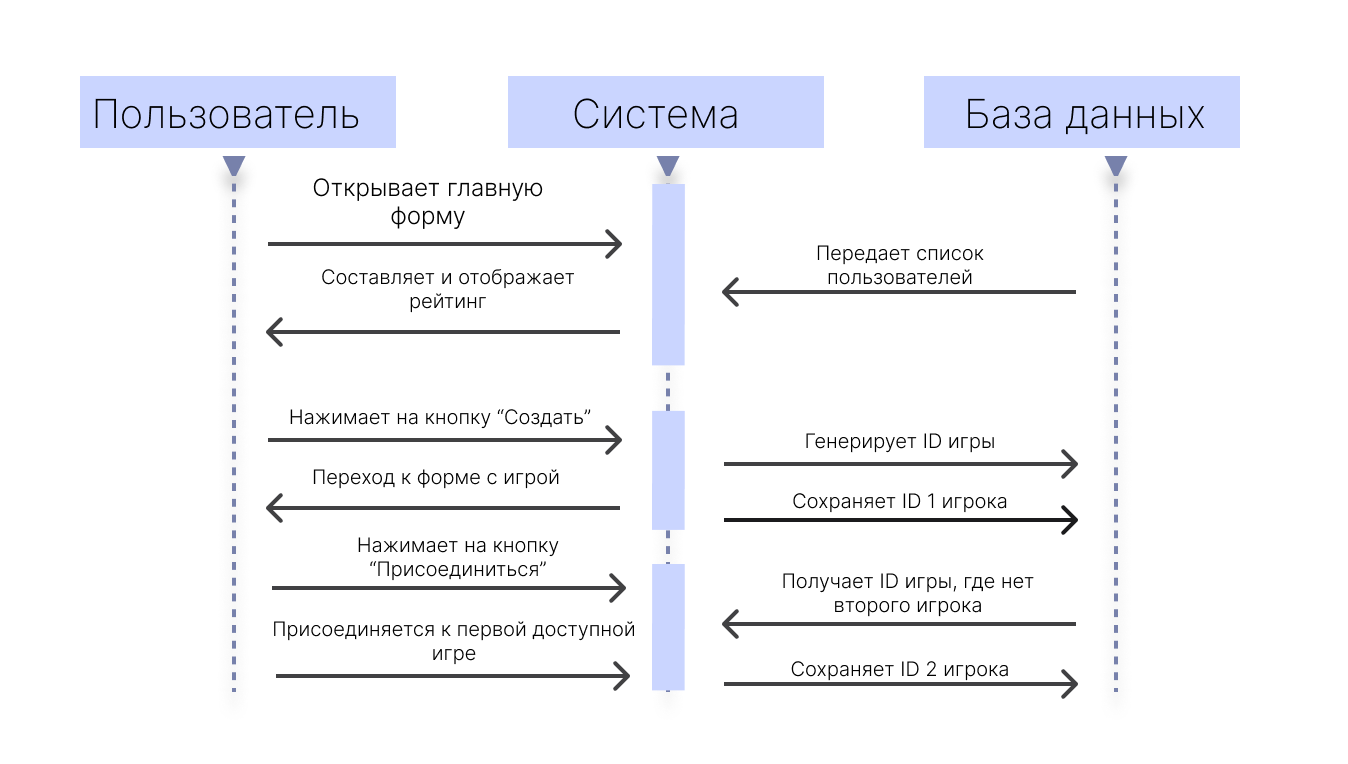
4.3 **Место:**

* Главная форма

4.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать дизайн формы входа |
| 2 | Добавить отображение рейтинга |
| 3 | Реализовать функционал создания игры |
| 4 | Реализовать функционал присоединения игры |
| 5 | Реализовать переход на форму профиля |
| 6 | Связать главную форму с формой входа |
| 7 | Реализовать обновление таблицы рейтинга |

* 1. **Диаграмма последовательностей**

****

* 1. **Техническое задание**

1. Реализовать дизайн формы

Основные элементы:

1. Подзаголовок

Тип элемента: текст

1. Кнопка «Профиль»

Функция: переход на форму профиля

Тип элемента: кнопка

Цвет: зеленый

1. Кнопка «Выход»

Функция: переход на форму входа

Тип элемента: кнопка

Цвет: оранжевый

1. Кнопка «Создать игру»

Функция: создание новой игры и переход на форму игры

Тип элемента: кнопка

Цвет: зеленый

1. Кнопка «Присоединиться к игре»

Функция: присоединиться к уже существующей игре

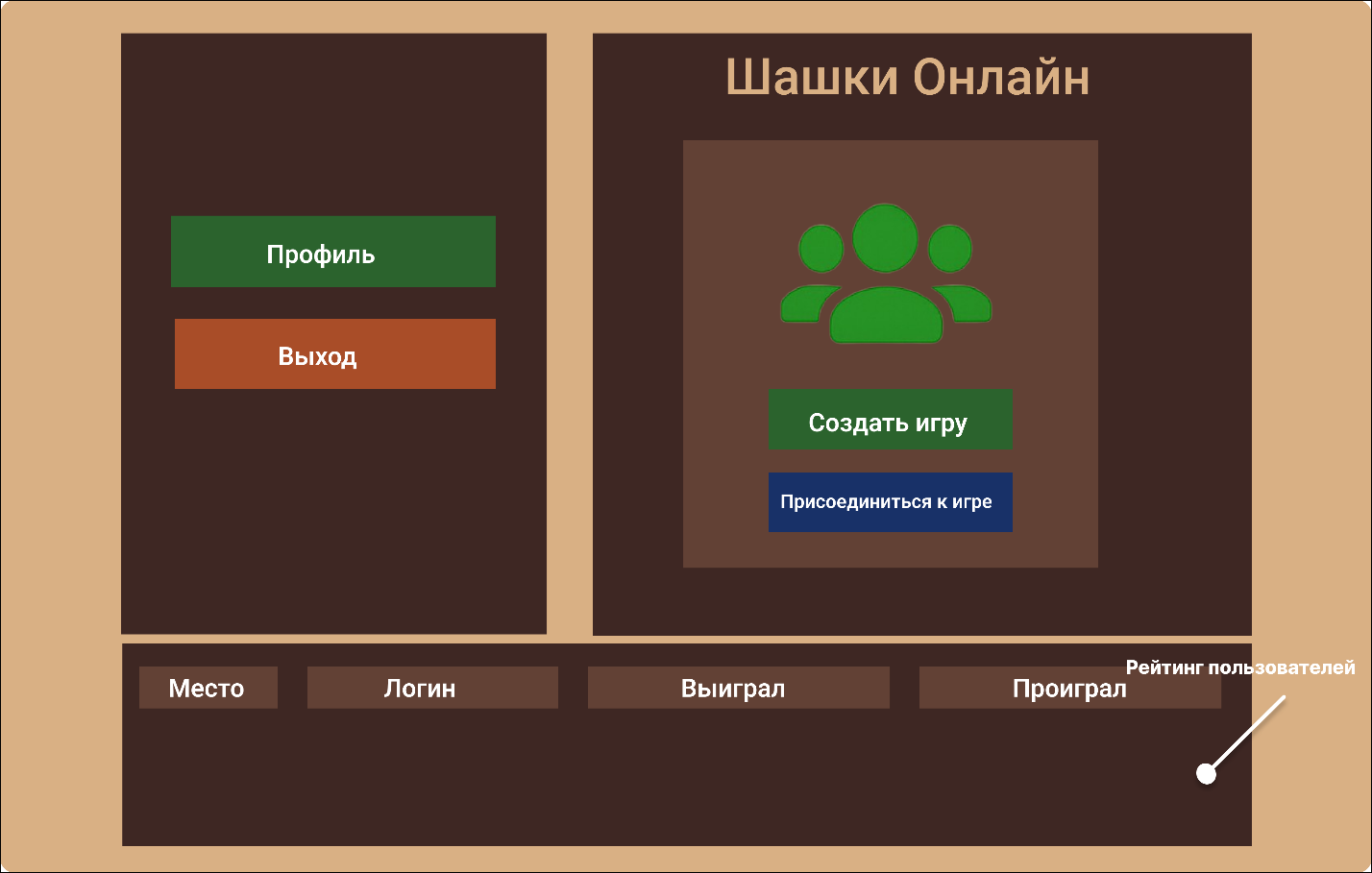
Тип элементов: кнопка

Цвет: синий

1. Таблица рейтинга

Функция: рейтинг пользователей

Тип элемента: список

Макет главной формы

1. Добавить отображение рейтинга

Рейтинг формируется на основе выигранных побед пользователей (чем больше побед, тем выше участник в рейтинге)

1. Вывести данные в рейтинге (место, логин, победы, поражения)
2. Реализовать функционал создания игры

При нажатии на кнопку «Создать игру», создается отдельная игра и пользователь переходит на форму игры. При этом пользователь, который создает игру, автоматически будет играть за белых.

1. Реализовать функционал присоединения игры

При нажатии на кнопку «Присоединиться к игре», пользователь переходит на форму с игрой, где ожидается второй участник. Такой пользователь будет играть за черных. Иначе вывести ошибку: «Нет доступных игр».

1. Реализовать переход на форму профиля
2. При нажатии на кнопку «Профиль», пользователь переходит на форму с профилем.
3. Связать главную форму с формой входа
4. При нажатии на кнопку «Выход» пользователь возвращается на форму входа.
5. Реализовать обновление таблицы рейтинга
6. Обновление рейтинга происходит после завершения игры.
7. Загрузить данные из БД, очистить текущую таблицу и перезаполнить её новыми данными.

5**.User Story**: Игра

Как зарегистрированный пользователь, я хочу иметь возможность сыграть с другим игроком.

5.2 **Пользователь:**

* **Игрок 1 (Белые) –** создает комнату и делает первый ход
* **Игрок 2 (Черные) –** присоединяется к игре и ожидает свой ход

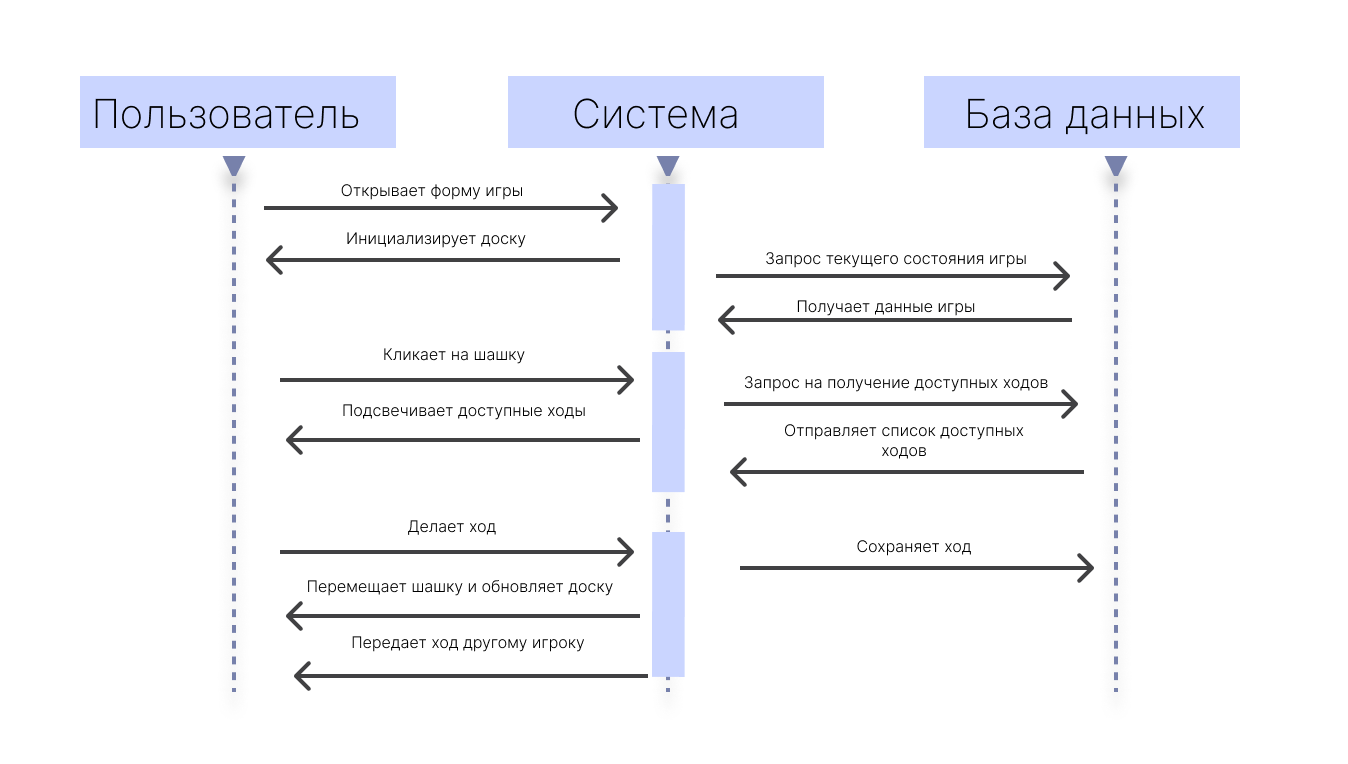
5.3 **Место:**

* Основная форма игры

5.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать дизайн формы входа |
| 2 | Реализовать функционал самой игры |
| 3 | Реализовать выхода из игры |
| 4 | Обновлять рейтинг после каждой игры |

* 1. **Диаграмма последовательностей**

****

* 1. **Техническое задание**

1. Реализовать дизайн формы

Основные элементы:

1. Доска и шашки
2. Кнопка «Завершить ход»

Функция: переход хода к следующему игроку

Тип элемента: кнопка

Цвет: зеленый

1. Кнопка «Сдаться»

Функция: закончить игру досрочно (проиграть)

Тип элемента: кнопка

Цвет: синий

1. Статус игроков

Функция: показать статус (ход и ожидание)

Тип элемента: панель

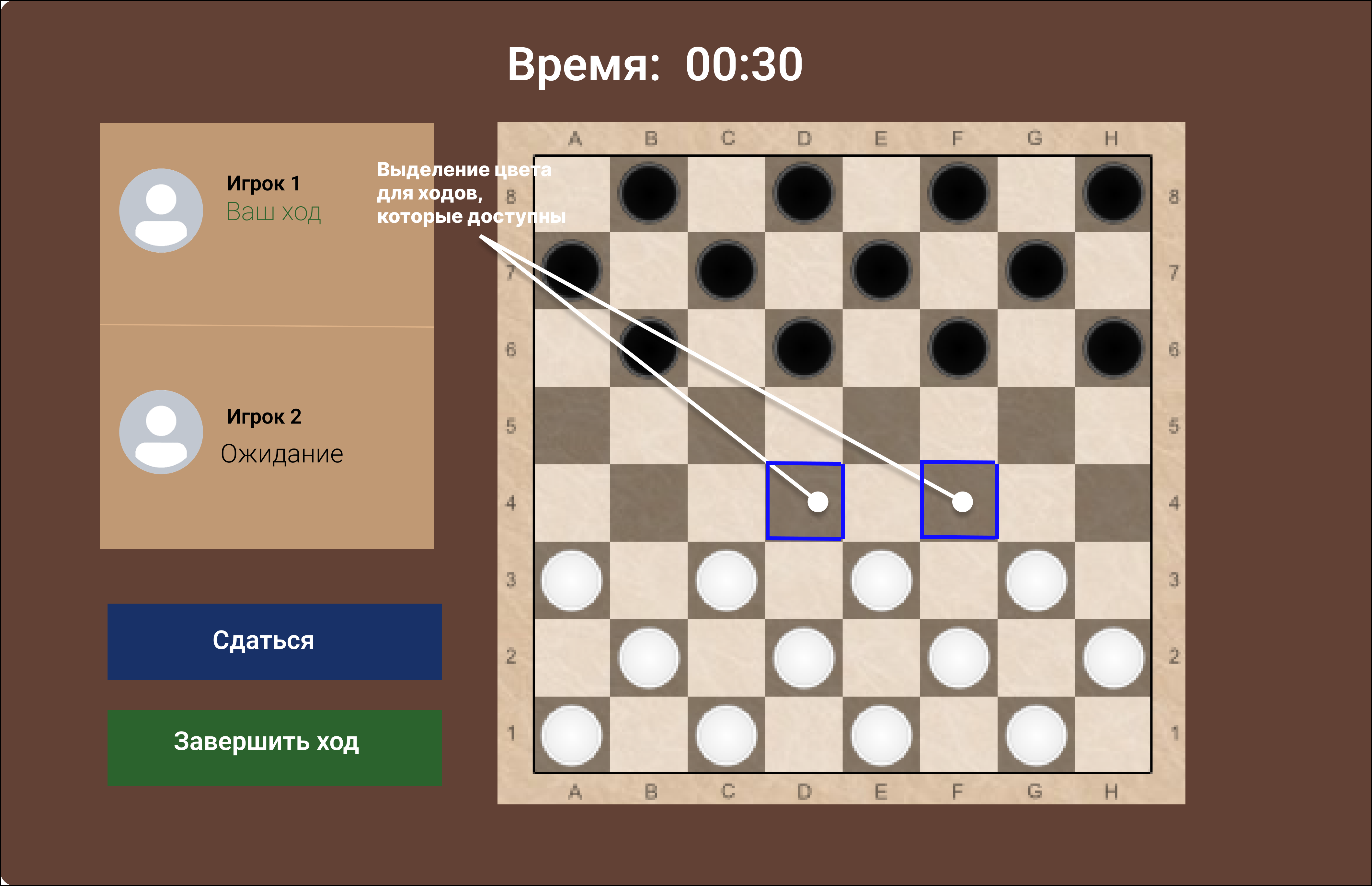
Цвет: бежевый

1. Иконки пользователей

Функция: показать фото профилей

Тип элемента: фото

Макет формы для игры



1. Реализовать функционал самой игры

Форма Игры — это форма, в котором пользователь взаимодействует с игровым полем, делает ходы и видит изменения от другого игрока. Форма должна:

1. Отобразить шашечную доску

При запуске игры, должна создаваться новая доска (8 x 8), шашки расставляются на первоначальные места. Включаются таймеры (30 секунд на ход и периодическая проверка изменений от другого игрока)

1. Предоставить обработку кликов по клеткам

Когда пользователь кликает мышью по доске, определяются координаты клетки. Если сейчас очередь этого игрока — система проверяет, что он выбрал свою шашку. После выбора шашки подсвечиваются все возможные ходы. Если пользователь выбирает одну из доступных позиций: шашка перемещается

ход передаётся другому игроку. Время хода сбрасывается

Новое состояние игры отрисовывается на доске. Если пользователь ошибся: выбор шашки отменяется, выводится сообщение о недопустимом ходе. Также если шашка достигает противоположного края доски: она превращается в дамку. Дамка может двигаться по диагонали на любое расстояние

1. Обеспечить передачу хода между игроками

Система должна следить за очередью:

Пользователь может сделать ход только в свой ход. Если попытка сделать ход вне своей очереди — выводится предупреждение

После завершения хода система автоматически передаёт право хода второму игроку. При переходе хода обновляется таймер, а также статус игрока («Ожидание» и «Ваш ход»)

1. Связать с базой данных для сохранения состояния игры

Каждый сделанный ход: сохраняется в базе данных через сервис системы.

В таблице хранятся:

ID игры

Цвет игрока

Исходная и конечная позиция

Время хода

1. Автоматически обновлять доску при новых ходах

Форма регулярно должна проверять наличие новых ходов от другого игрока: происходит запрос к серверу. Если были новые ходы, доска должна перестраиваться заново. Все изменения будут отражаться на экране

Это позволит пользователю видеть актуальное состояние игры, даже если он был в ожидании.

1. Получить данные пользователя (логин, игры, победы, поражения)
2. Реализовать выход из игры

Игра заканчивается, когда один из игроков сдался или у одного из игроков не осталось шашек

В этот момент:

Игровое поле очищается

Пользователь должен переходить обратно в главное меню

1. Обновлять рейтинг после каждой игры

При завершении игры:

Должна обновляться статистика в базе данных:

Увеличиваться количество побед у победителя

Увеличиваться количество поражений у проигравшего

Система снова формирует рейтинг пользователя

6**.User Story**: Управление игровыми сессиями

Как активный игровой пользователь, я хочу иметь возможность управлять игрой (сдаться и завершить ход).

6.2 **Пользователь:**

* **Активный игрок** может либо сдаться, либо завершить ход

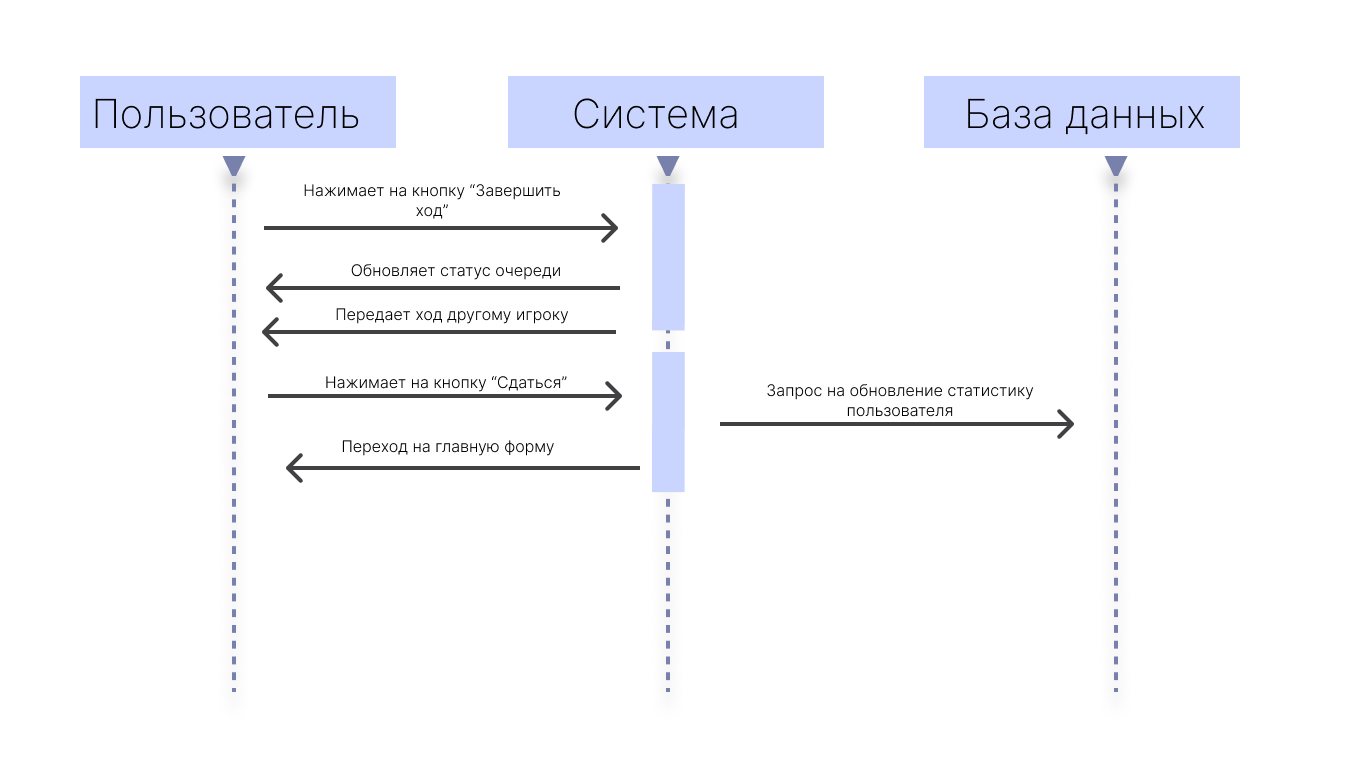
6.3 **Место:**

* Форма Игры

6.4 **Задачи**

|  |  |
| --- | --- |
| **№** | **Задача** |
| 1 | Реализовать ручное завершение формы |
| 2 | Реализовать функционал кнопки «Сдаться» |
| 3 | Обновить статистику в БД |
| 4 | Связать форму игры с главной формой |

* 1. **Диаграмма последовательностей**

****

* 1. **Техническое задание**

1. Реализовать ручное завершение формы

Игрок кликает на кнопку «Завершить ход». Сервис должен обнулить доступные ходы, передать очередь другому игроку, обновить доску, ход должен сохраниться в БД.

1. Реализовать функционал кнопки «Сдаться»

При нажатии на кнопку «Сдаться» игра должна завершаться, противник оглашается победителем и обновляется в БД статистика пользователей.

1. Обновить статистику в БД

При завершении игры досрочно, статистика должна обновляться (у победителя – увеличивается количество побед, у проигравшего количество проигрышей)

1. Связать форму игры с главной формой

После завершения игры пользователь должен переходить на главную форму

* 1. Требования к логической структуре БД

*Хранение данных:* Все данные должны храниться в реляционной базе данных (БД) с соответствующими таблицами (например, таблица "Users") с полями:

|  |  |  |
| --- | --- | --- |
| Поле: | Тип данных: | Описание: |
| Id | UUID | Уникальный идентификатор пользователя. |
| Login | TEXT | Логин пользователя. |
| PasswordHash | TEXT | Хэшированный пароль. |
| GamesPlayed | INTEGER | Количество сыгранных игр. |
| Wins | INTEGER | Количество побед. |
| Losses | INTEGER | Количество проигрышей |

“Moves”:

|  |  |  |
| --- | --- | --- |
| Поле: | Тип данных: | Описание: |
| Id | UUID | Уникальный идентификатор хода. |
| GameId | UUID | Уникальный идентификатор игры, к которой относится ход. |
| MoveNumber | INTEGER | Номер хода. |
| PlayerColor | TEXT | Цвет игрока, сделавшего ход. |
| FromPosition | TEXT | Позиция, откуда был сделан ход. |
| ToPosition | TEXT | Позиция, куда был сделан ход. |
| CreatedAt | TIMESTAMPTZ | Время создания хода. |

“GameHistories”:

|  |  |  |
| --- | --- | --- |
| Поле: | Тип данных: | Описание: |
| Id | UUID | Уникальный идентификатор истории игр. |
| GameId | UUID | Уникальный идентификатор игры, к которой относится запись. |
| IsWin | BOOLEAN | Указывает, выиграл ли пользователь игру. |
| PlayedAt | TIMESTAMPTZ | Дата и время, когда игра была завершена. |
| UserId | UUID | Уникальный идентификатор пользователя участвовавшего в игре |

“Games”:

|  |  |  |
| --- | --- | --- |
| Поле: | Тип данных: | Описание: |
| Id | UUID | Уникальный идентификатор игры. |
| WhitePlayerId | UUID | Уникальный идентификатор пользователя за белых. |
| BlackPlayerId | UUID | Уникальный идентификатор пользователя за черных. |
| StartedAt | TIMESTAMPTZ | Время начала игры. |
| FinishedAt | TIMESTAMPTZ | Время окончания игры. |
| Winner | TEXT | Победитель игры. |
| Status | TEXT | Статус игры |