Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Казанский (Приволжский) федеральный университет»

*Институт вычислительной математики и информационных технологий*

**Техническое задание**

Игровое приложение «Домино»

Выполнили

обучающиеся 1 курса факультета «Прикладной математики» Балобанова Арина, Лисюткина Руслана, Назарова Дарья

Казань, 2025

СОДЕРЖАНИЕ

[1. Обзор системы 3](#_Toc200034906)

[1.1 Обзор вариантов пользователей 3](#_Toc200034907)

[1.2. Нефункциональные требования 17](#_Toc200034908)

[1.3. Требования к логической структуре БД 18](#_Toc200034909)

# Обзор системы

## 1.1 Обзор вариантов пользователей

**1. User Story 1**. Регистрация нового пользователя:

Новый пользователь хочет иметь возможность зарегистрироваться в системе, создав учетную запись с уникальным логином и паролем.

**2. Пользователи:**

- новый пользователь:

• Незарегистрированный пользователь

**3. Места:**

- форма регистрации

**4. Задачи:**

- Создать форму регистрации с полями логина, пароля и подтверждения пароля

- Реализовать валидацию введенных данных

- Отобразить сообщение об ошибке при некорректных данных

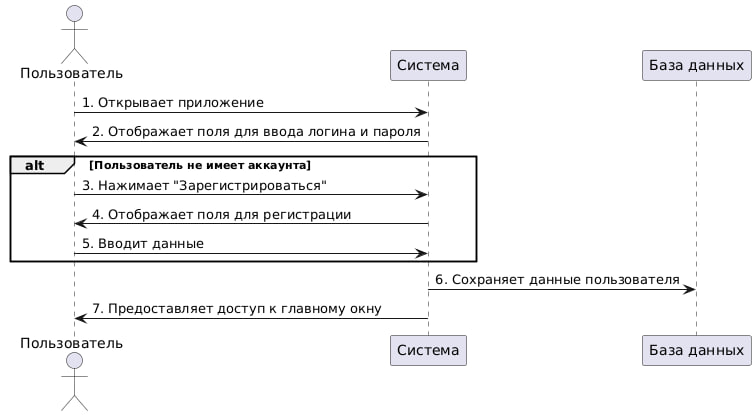
- Проверить уникальность логина в системе

- Реализовать хэширование введенного пароля

- Сохранить данные нового пользователя в БД

- Перенаправить на главное окно после успешной регистрации

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Создать форму регистрации

- Создать Windows Form с полями для логина, пароля и подтверждения пароля

- Добавить кнопку "Регистрация"

- Реализовать просмотр введенного пароля

- Реализовать скрытие пароля спец символом \*

Задача 2: Реализовать валидацию

- Проверить совпадение введенных паролей

- Проверить заполненность полей

- Проверить корректность логина и пароля ( логин от 2 до 20 символов, только буквы и цифры, пароль должен содержать минимум 8 символов, латинские буквы, хотя бы одну цифру и заглавную букву).

- Проверить уникальность логина в БД

- При ошибках валидации показывать соответствующие сообщения

Задача 3: Сохранить пользователя

- Создать запись в таблице Users

- Хэшировать пароль перед сохранением

- Инициализировать начальный рейтинг пользователя

- Обрабатывать возможные исключения, такие как ошибки подключения к базе данных, ошибки при выполнении запроса.

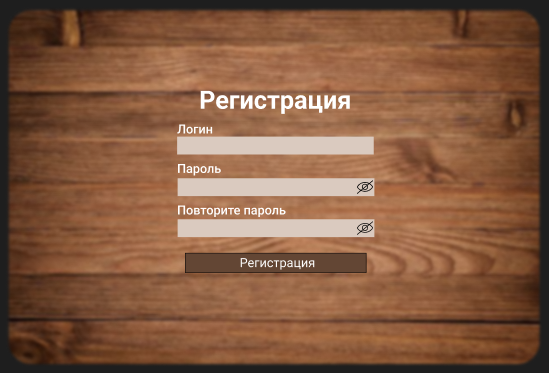
Задача 4: Обеспечить навигацию

- При успешной регистрации открывать главное окно игры

- Передать логин зарегистрированного пользователя в главное окно

- При ошибках регистрации показывать соответствующие сообщения

Макет окна регистрации:



**1. User Story 2.** Авторизация существующего пользователя:

Зарегистрированный пользователь хочет иметь возможность войти в систему под своими учетными данными для доступа к игровому функционалу.

**2. Пользователи:**

- зарегистрированный пользователь:

• Существующий пользователь системы

**3. Места:**

- форма авторизации

**4. Задачи:**

- Создать форму авторизации с полями логина и пароля

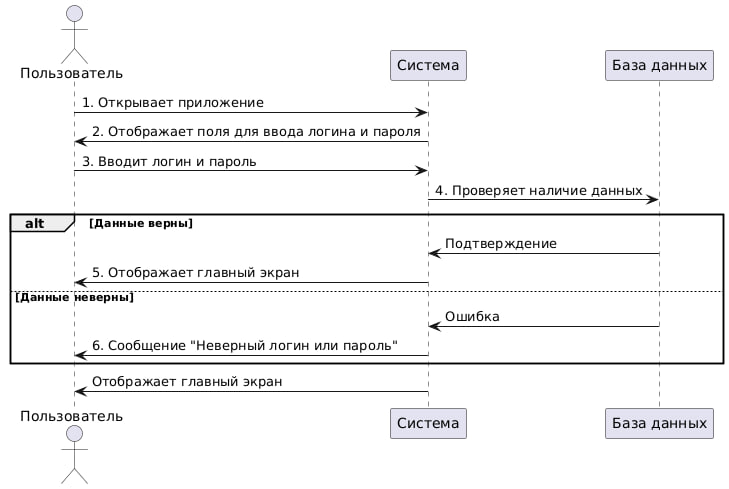
- Реализовать проверку введенных данных с данными зарегистрированного пользователя, лежащие в БД

- Обеспечить безопасную проверку паролей (хеширование)

- Отобразить сообщение об ошибке при неверных данных

- Перенаправить на главное окно при успешной авторизации

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Создать форму авторизации

- Создать форму с полями для логина и пароля

- Добавить текст "Нет аккаунта?"

- Добавить кнопки "Вход" и "Регистрация"

- Добавить переключатель языка (Русский/English)

- Добавить скрытие пароля спец символом \*

- Добавить локализацию

Задача 2: Реализовать проверку данных

- Создать метод для проверки логина и пароля с данными, лежащими в БД

- Проверить заполненность полей

- Реализовать хеширование паролей

- Реализовать просмотр введенного пароля

- Отобразить сообщение об ошибке при неверных данных

- Отобразить сообщение об ошибке при незаполненных полях

Задача 3: Обеспечить навигацию

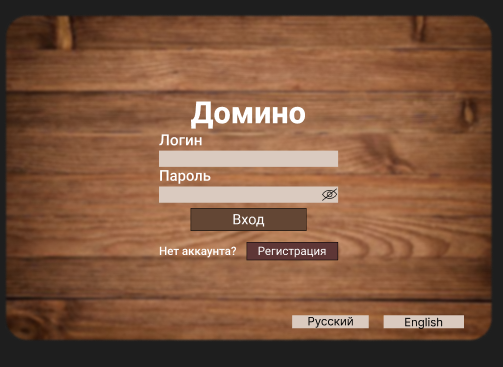
- При успешной авторизации открывать главное окно игры

- При неуспешной авторизации показывать сообщение об ошибке

- При нажатии на кнопку "Регистрация" открывать форму регистрации

Макет окна авторизации:

Окно авторизации – первоначальное окно, которое видит пользователь при загрузке приложения. На окне авторизации отображены поля для ввода логина и пароля. Зарегистрированный пользователь вводит свои данные и нажимает на кнопку «Вход». В случае незарегистрированного пользователя нужно воспользоваться кнопкой «Регистрация».



**1. User Story 3**. Просмотр рейтинга очков пользователей:

Пользователь хочет иметь возможность просматривать рейтинг всех игроков системы, отсортированный по количеству очков.

**2. Пользователи:**

- пользователь:

• Любой зарегистрированный игрок

**3. Места:**

- панель рейтинга в главном окне

**4. Задачи:**

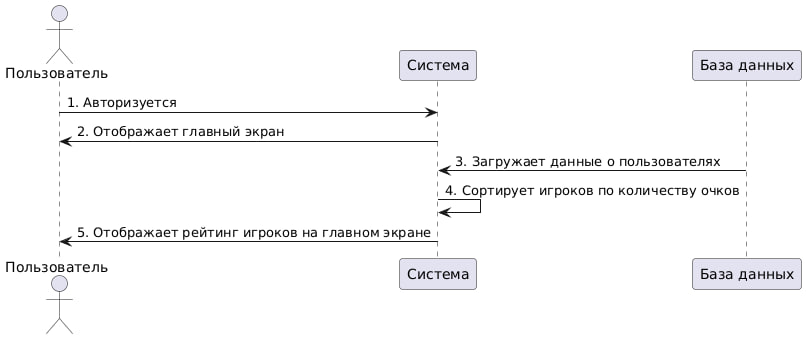
- Создать компонент для отображения рейтинга игроков

- Реализовать загрузку данных о всех пользователях из БД

- Отобразить логин и количество очков каждого игрока

- Обновлять рейтинг при изменении очков игроков

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Создать компонент рейтинга

- Создать таблицу для отображения рейтинга в главном окне

- Добавить колонки "Игрок" и "Очки"

- Добавить название таблицы "Рейтинг"

Задача 2: Реализовать загрузку данных

- Получить данные о пользователях из БД

- Сортировать данные по количеству очков в порядке убывания

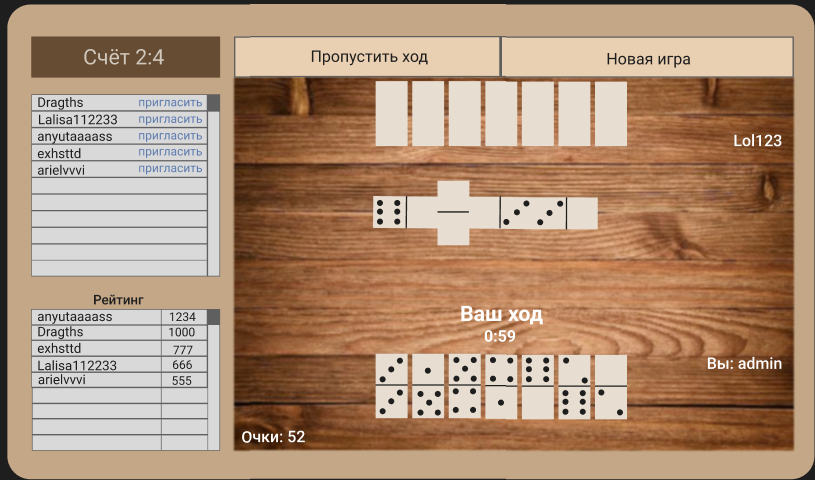
- Отображать данные пользователя (логин, пароль)

Задача 3: Обеспечить обновление данных

- Реализовать механизм обновления рейтинга после завершения игр

- Обновлять отображение при изменении данных

Макет главного окна(просмотр рейтинга-левый нижний угол)



**1. User Story 4.** Просмотр списка доступных игроков:

Пользователь хочет иметь возможность видеть список всех онлайн игроков, которых можно пригласить к игре.

**2. Пользователи:**

- пользователь:

• Игрок, желающий найти соперника

**3. Места:**

- панель списка игроков в главном окне

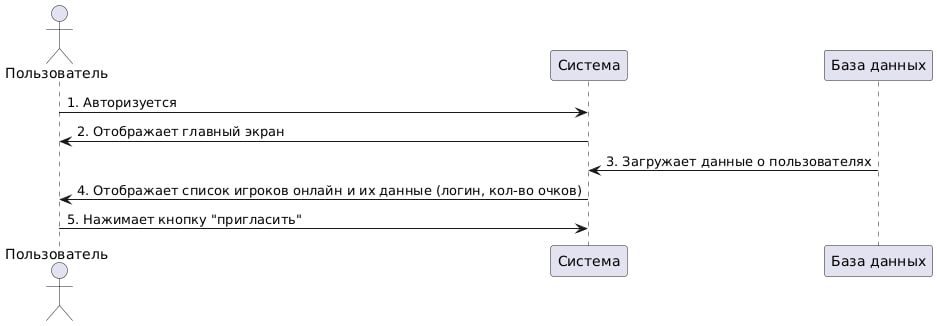
**4. Задачи:**

- Создать компонент для отображения списка доступных игроков

- Добавить возможность приглашения игроков из списка

- Обновлять список в реальном времени

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Создать компонент списка игроков

- Создать таблицу для отображения списка игроков в главном окне

- Получить логин игроков из БД

- Добавить напротив логина кнопку «пригласить»

Задача 2: Обеспечить функционал приглашения

- Получить UserId выбранного игрока

- Создать метод для отправки приглашения

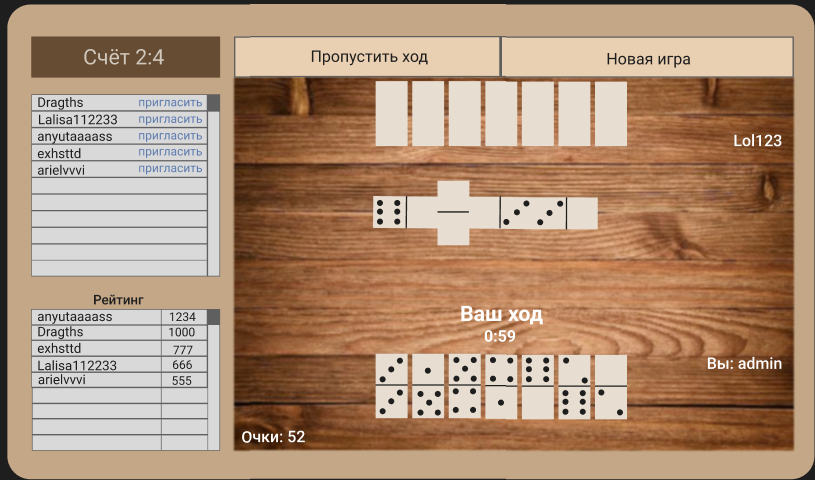
- После успешной отправки изменить текст кнопки на "отправлено"

- Изменить цвет строки на светло-серый для визуального отличия

- Отобразить уведомление "Приглашение отправлено игроку [Логин]"

Задача 3: Обеспечить своевременное обновление списка доступных пользователей

Макет главного экрана (левый верхний угол- просмотр игроков, доступных для приглашения)



**1. User Story 5.** Игра в Домино с другими пользователями:

Пользователь хочет иметь возможность играть в домино с другим игроком, соблюдая правила игры и делая ходы по очереди.

1. **Пользователи:**

- игрок, делающий ход:

• Игрок 1

- игрок, ожидающий хода:

• Игрок 2

**3. Места:**

- игровое поле в главном окне

**4. Задачи:**

- Реализовать игровое поле для размещения костяшек

- Отобразить костяшки домино каждого игрока

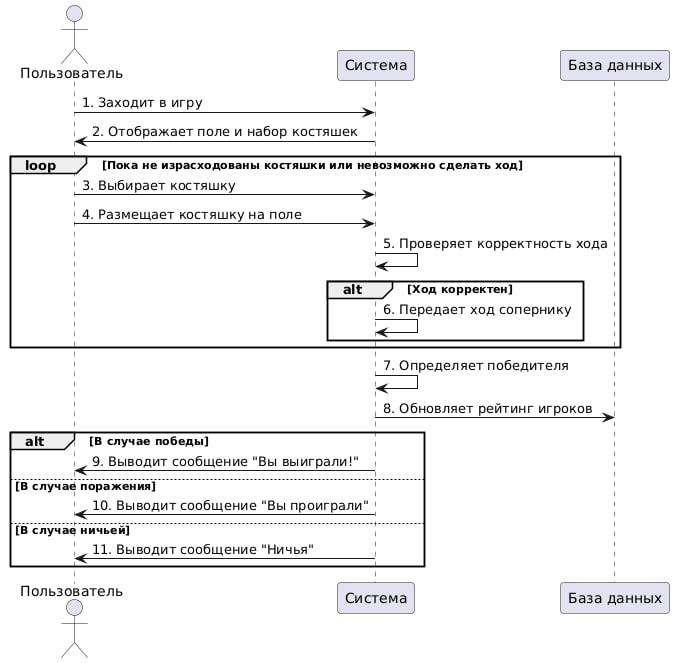
- Реализовать логику размещения костяшек на игровом поле

- Проверять корректность ходов согласно правилам домино

- Передавать ход между игроками

- Определять победителя или ничью

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Реализовать игровое поле для размещения костяшек

- Реализовать возможность перетаскивания костяшек из руки игрока на поле

- Создать метод для определения координат ячейки

- Реализовать метод для проверки свободности ячейки

Задача 2: Отобразить костяшки домино каждого игрока

- Реализовать визуальное отображение костяшки в виде прямоугольника

- Создать панель для отображения костяшек игрока

- Разместить до 7 костяшек в ряд

- Добавить возможность выбора костяшки

Задача 3: Реализовать логику размещения костяшек на игровом поле

- Для первого хода разрешить размещение костяшки в любой ячейке поля

- Для последующих ходов проверять, что костяшка размещается рядом с уже размещенными

- Проверить совпадение значений на соприкасающихся сторонах костяшек

- Реализовать метод для размещения костяшки на поле

- Обновлять визуальное отображение поля после размещения костяшки

- Удалять размещенную костяшку игрока

Задача 4: Проверять корректность ходов согласно правилам домино

- Создать метод для проверки хода

- Проверить, что ячейка для размещения свободна

- Для первого хода: разрешить размещение в любом месте поля

- Для последующих ходов: проверить наличие соседних костяшек

- При некорректном ходе отобразить сообщение об ошибке

- Вернуть костяшку игроку при неудачном размещении

Задача 5: Передавать ход между игроками

- Создать метод для смены активного игрока

- Обновить интерфейс: отобразить "Ваш ход" или "Ход соперника"

- Заблокировать возможность хода для неактивного игрока

- Добавить таймер для хода (1 минута)

- Обновить счетчик ходов и отобразить в интерфейсе

Задача 6: Определять победителя или ничью

- Проверить, закончились ли костяшки у одного из игроков (победа)

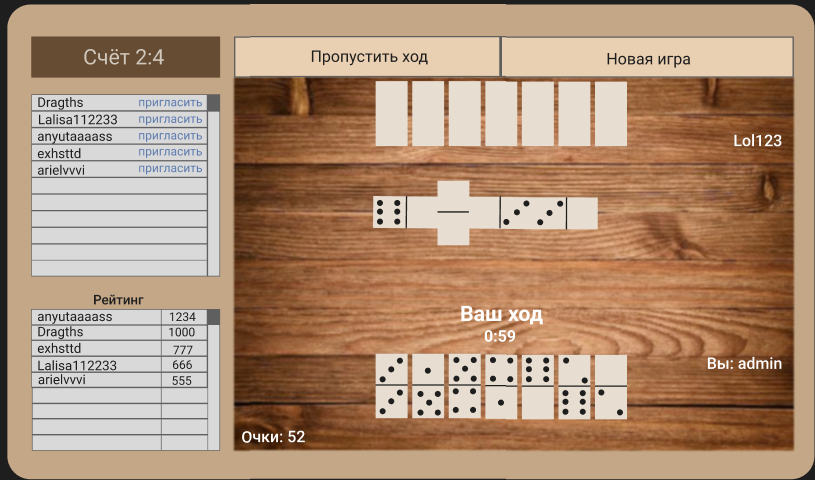
- Проверить, могут ли игроки сделать ход (если нет - подсчет очков)

- Создать метод для подсчета очков оставшихся костяшек

- Определить победителя: меньше очков = победа, равные очки = ничья

- Отобразить результат игры в окне «Счёт»

Меню главного окна (Игровое поле чуть правее середины)



**1. User Story** 6. Просмотр и управление приглашениями от других игроков:

Как пользователь, я хочу иметь возможность просматривать входящие приглашения к игре и принимать или отклонять их.

**2. Пользователи:**

- получатель приглашения:

• Игрок, которого приглашают

- отправитель приглашения:

• Игрок, отправивший приглашение

**3. Места:**

- окно приглашений

**4. Задачи:**

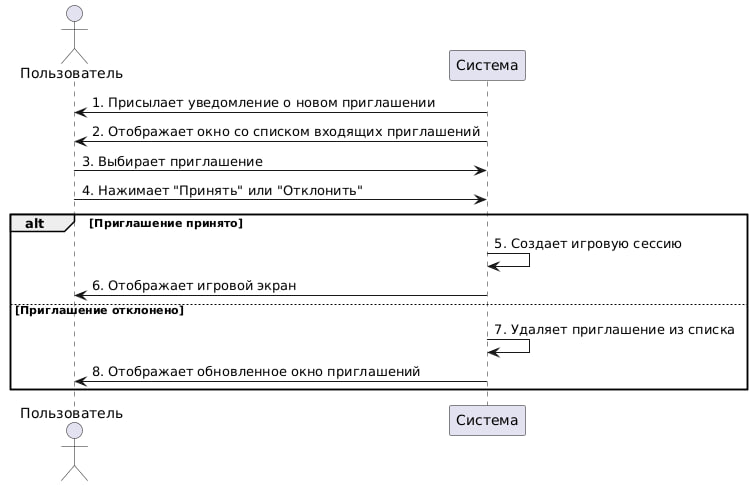
- Создать окно для отображения списка входящих приглашений

- Отобразить информацию о приглашающих игроках (логин, очки)

- Добавить кнопки "Принять" и "Отклонить" для каждого приглашения

- Реализовать логику принятия приглашения и начала игры

**5. Диаграмма последовательности**



**6. Техническое задание**

Задача 1: Создать окно приглашений

- Добавить заголовок "Приглашения"

- Настроить колонки: "Логин", "Очки"

- Добавить две кнопки: "Принять" и "Отклонить"

- Создать компонент для отображения списка приглашений

- Отображать логин и рейтинг приглашающего игрока

Задача 2: Реализовать обработку приглашений

- Создать метод для загрузки приглашений

- Создать метод для принятия приглашения

- Закрыть окно приглашений

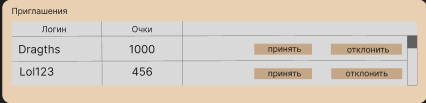
- Открыть игровой интерфейс для обоих игроков

- Создать метод для отклонения

- Уведомить отправителя об отклонении

-Закрывать окно автоматически, если все приглашения обработаны (список пуст)

Макет окна приглашений



**1. User Story 7.** Управление игровыми сессиями:

Как игрок в активной игре, я хочу иметь возможность управлять текущей игровой сессией: пропустить ход или начинать новую игру.

**2. Пользователи:**

- активный игрок:

• Игрок, участвующий в текущей игре

**3. Места:**

- панель управления игрой в главном окне

**4. Задачи:**

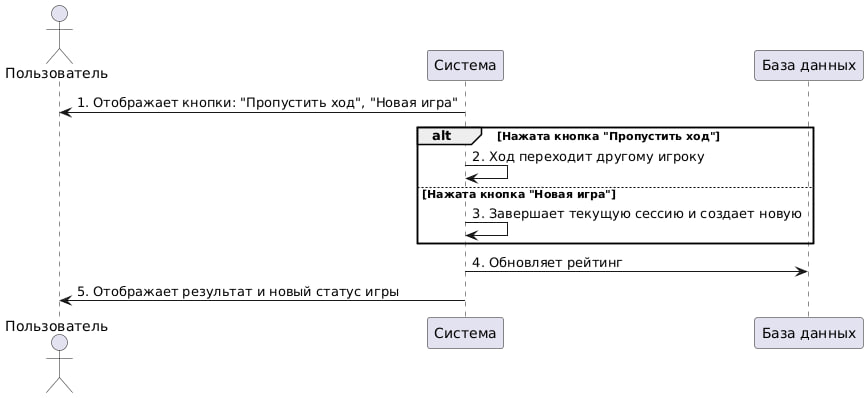
- Добавить кнопки управления игрой: "Пропустить ход", "Новая игра"

- Реализовать логику предложения пропуска хода (переход хода к сопернику)

- Реализовать логику начала новой игры (сброс игрового поля)

- Обновлять очки игроков по результатам игры

**5. Диаграмма последовательности**



**6. Техническое задание:**

Задача 1: Добавить элементы управления игрой

- Создать кнопки " Пропустить ход ", "Новая игра"

- Разместить кнопки в интерфейсе игрового окна

- Настроить обработчики событий для кнопок

Задача 2: Реализовать логику пропуска хода

- Создать метод для перехода хода к сопернику

- Обновлять интерфейс для обоих игроков

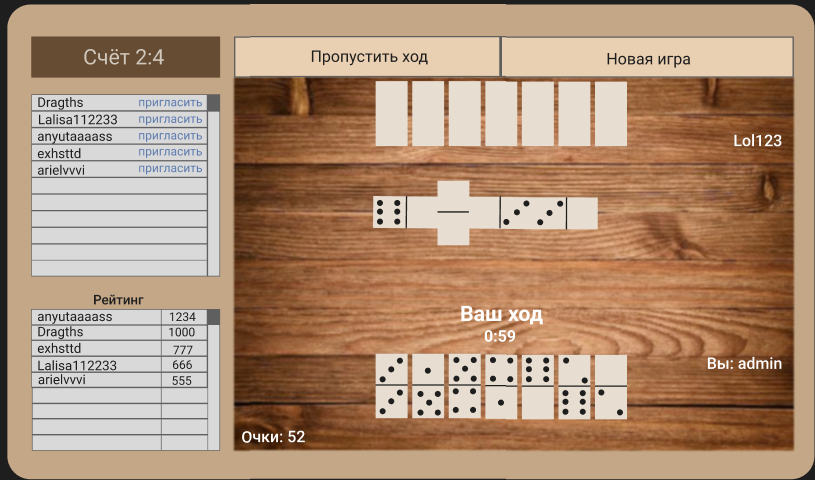
Задача 4: Реализовать логику новой игры

- Создать метод для завершения текущей игры и создания новой

- Сбрасывать игровое поле и перераспределять костяшки

- Обновлять интерфейс для обоих игроков

Макет главного окна(верхние кнопки игрового поля)



## 1.2. Нефункциональные требования

Приложение написано на языке программирования C#.

Используемые инструменты: Visual Studio, Windows Forms, .NET.

Приложение работает с реляционной базой данных PostgreSQL и СУБД SQL.

## 1.3. Требования к логической структуре БД

1. Таблица Users

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| ID | UUID | Уникальный идентификатор |
| Login | Text | Логин пользователя |
| Password | Text | Пароль пользователя |
| Rating | Integer | Рейтинг пользователя |