Milestone 2 Report

# Team

SoPra Group 12

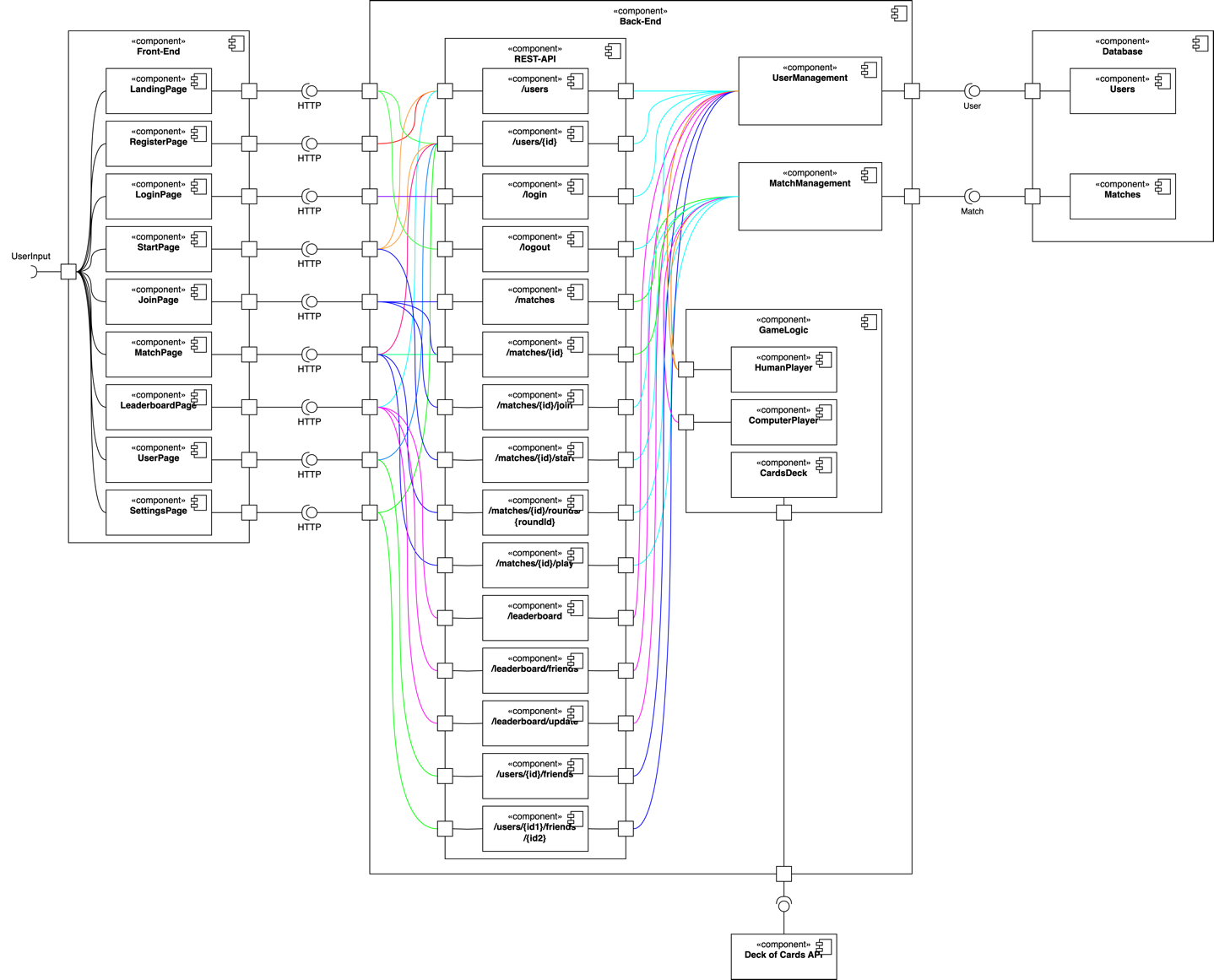
**Group Leader:** Borislav Rakic

**Members:**

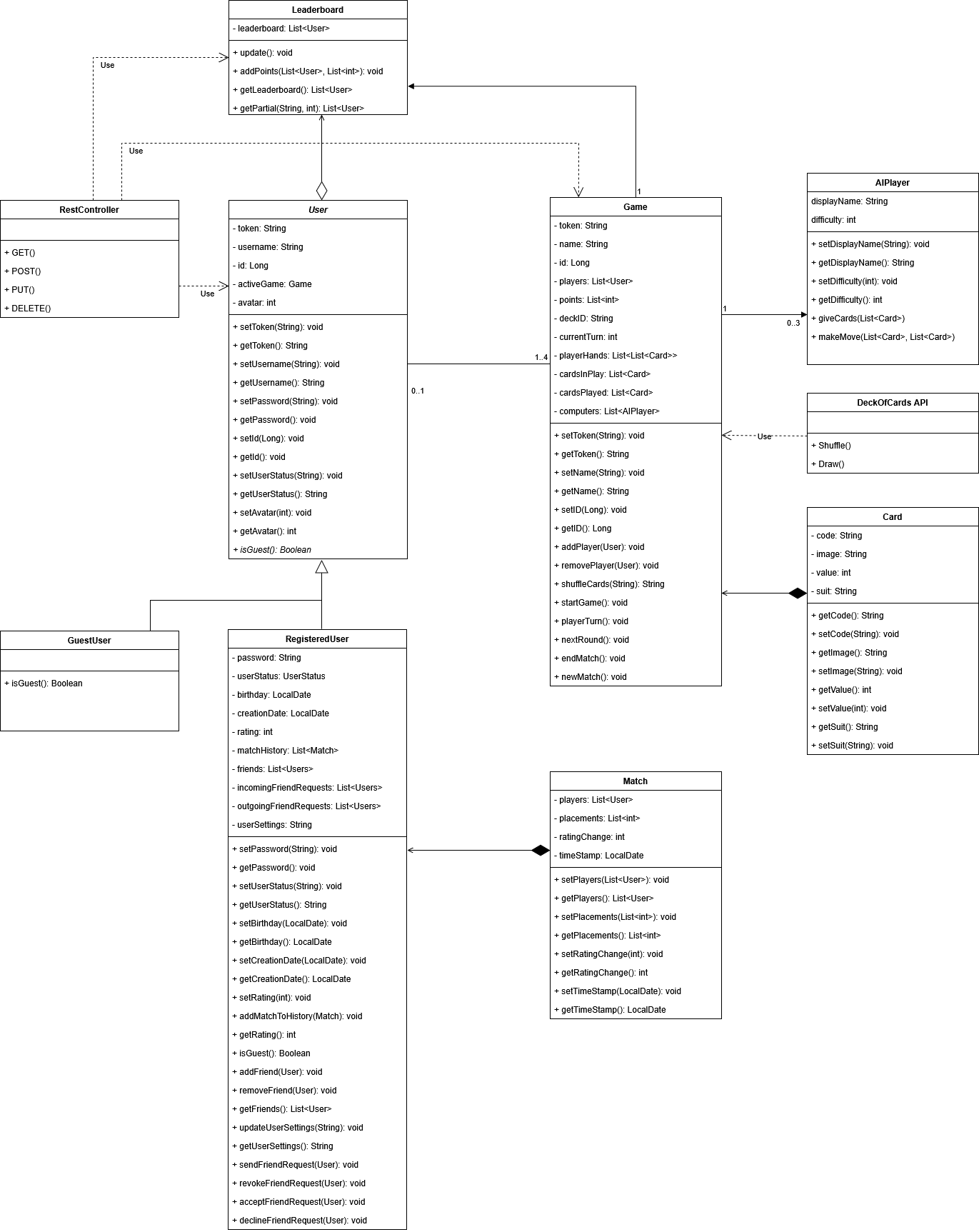
* Stevania Eilyn Frutiger
  + [stevaniaeilyn.frutiger@uzh.ch](mailto:stevaniaeilyn.frutiger@uzh.ch)
  + 23-731-748
  + GitHub: stevaniaeilyn
* Dominique Heller
  + [dominiqueoliver.heller@uzh.ch](mailto:dominiqueoliver.heller@uzh.ch)
  + 17-918-566
  + GitHub: dominiqueheller
* Borislav Rakic
  + [borislav.rakic@uzh.ch](mailto:borislav.rakic@uzh.ch)
  + 21-923-610
  + GitHub: borislav-rakic
* Dieter Andreas Studer
  + [dieter.studer@access.uzh.ch](mailto:dieter.studer@access.uzh.ch)
  + 00-719-823
  + GitHub: diderot5038

# UML Diagrams

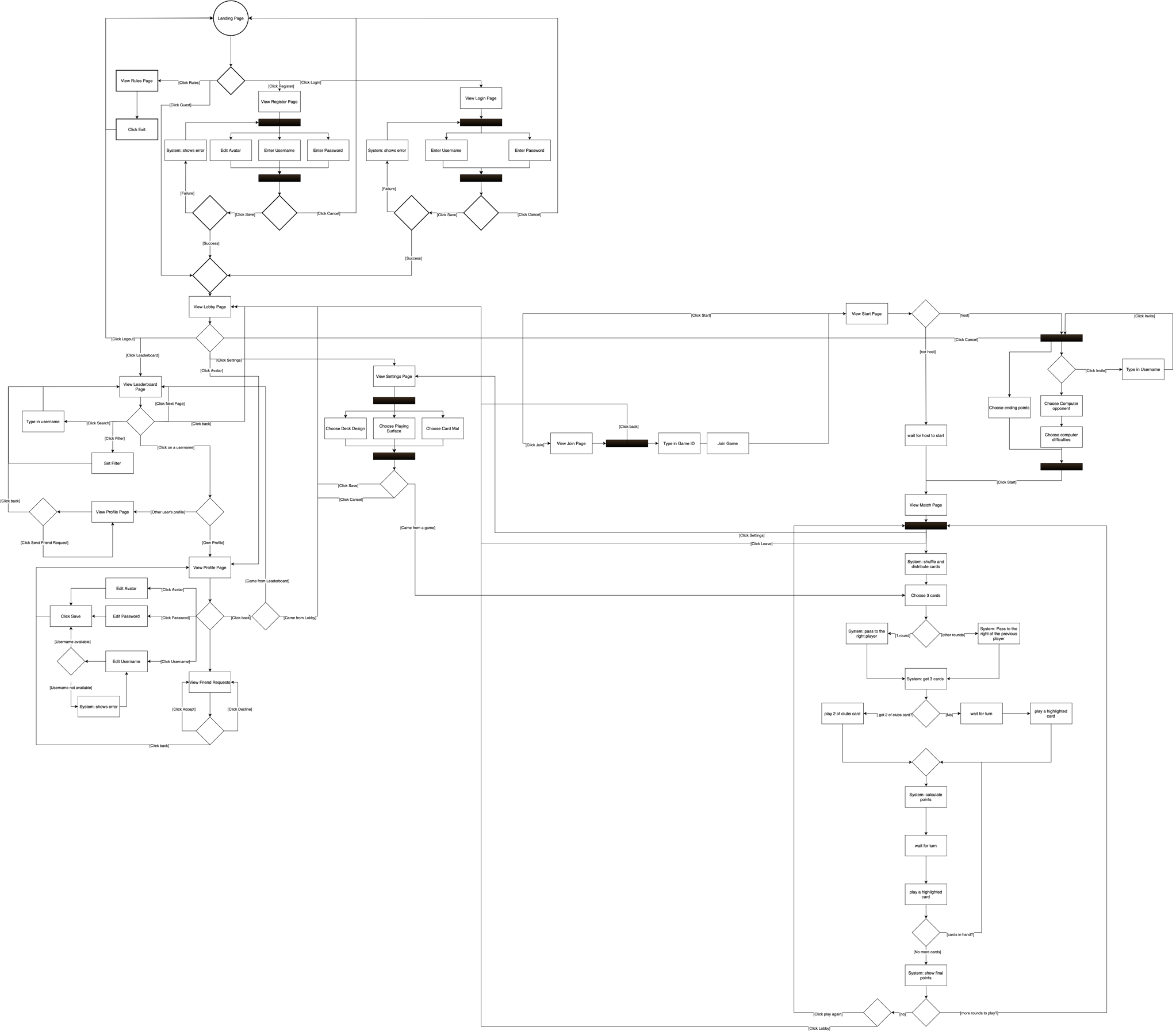
## Component Diagram



## Class Diagram



## Activity Diagram



# REST Specification

## Overview of Access Points

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mapping** | **Method** | **Request / DTO** | **Response / DTO** | **Status Code** | **Description** |
| /users | POST | UserCreateDTO | UserDTO  ErrorResponseDTO | 201 Created  409 Conflict | Successfully added user.  Failed:username exists. |
| /users/{userId} | GET | Long userId | UserDTO  ErrorResponseDTO | 200 OK  404 Not Found | Retrieve user profile with *userId.*  User with *userId* was not found. |
| /users/{userId} | PUT | UserUpdateDTO | UserDTO  ErrorResponseDTO | 200 OK  404 Not Found | Updated user profile.  User with *userId* was not found. |
| /login | POST | LoginRequestDTO | LoginResponseDTO  ErrorResponseDTO | 200 OK  401 Unauthorized | Successful login.  Error: Faulty credentials. |
| /logout | POST |  | 204 No Content  ErrorResponseDTO | 204 No Content  401 Unauthorized | Successful logout.  Denied logout: token was not ok. |
| /users/{userId1}/friends | GET | Long userId1 Long userId2 | List <FriendDTO>  ErrorResponseDTO | 200 OK  401 Unauthorized | Get list of friends.  User can only see their own friends. |
| /users/{userId1}/friends/{userId2} | POST | Long userId1 Long userId2 | 204 No Content  ErrorResponseDTO | 204 No Content  404 Not Found | Friend added.  Found no User with userId2. |
| /users/{userId1}/friends/{userId2} | DEL | Long userId1 Long userId2 | 204 No Content  ErrorResponseDTO | 204 No Content  404 Not Found | Friend removed.  Found no User with userId2. |
| /matches | POST | MatchCreateDTO | MatchDTO  ErrorResponseDTO | 201 Created  401 Unauthorized. | Successfully created match.  Error: Faulty token. |
| /matches/{matchId} | GET | Long matchId | MatchDTO  ErrorResponseDTO | 200 OK  404 Not Found | Get match details.  No match with matchID found. |
| /matches/{matchId}/join | POST | PlayerDTO | MatchDTO  ErrorResponseDTO | 200 OK  400 Bad Request | Seat reserved on match.  Joining not possible (e.g. overbooked). |
| /matches/{matchId}/start | POST | [GameOwnerOnly] | MatchDTO  ErrorResponseDTO | 200 OK  403 Forbidden | Start the game.  Not all Players ready. |
| /matches/{matchId}/rounds/{roundId} | GET | Long matchID, int roundId | RoundDTO  ErrorResponseDTO | 200 OK  404 Not Found | Get stats on round.  No such round played, yet. |
| /matches/{matchId}/play/ | POST | PlayCardDTO | PlayedCardDTO ErrorResponseDTO ErrorResponseDTO | 200 OK  404 Not Found  400 Bad Request | Card accepted in this round.  No such game or round at play.  Illegal card played. |
| /leaderboard/ | GET | n/a | LeaderBDTO | 200 OK | Return leaderboard data. |
| /leaderboard/friends | GET | n/a | LeaderBDTO | 200 OK  401 Unauthorized | Return leaderboard data of friends only.  Only logged in users can use this feature. |
| /leaderboard/update | POST | LeaderBUpdateDTO | 204 No Content  ErrorResponseDTO  ErrorResponseDTO | 204 No Content  404 Not Found  400 Bad Request | Score of listed players was updated.  This matchId/matchToken not found.  Mismatch of match and players. |

## DTOs

We are making use of the following DTOs:

### User Management

UserCreateDTO (String username, String password)

UserDTO (Long id, String username, String userstatus, LocalDate birthday, LocalDate creationDate, int rating, List<UserDTO> friends, String userSettings, Bool isGuest)

UserUpdateDTO (String username, String password LocalDate birthday, String userSettings)

LoginRequestDTO (String username, String password)

LoginResponseDTO (String token, Long userId, String username)

ErrorResponseDTO (String error, String message)

### Match Management

MatchCreateDTO (Long matchId, String name, List<PlayerDTO> players)

MatchDTO (Long id, List<PlayerDTO> players, Bool started)

RoundDTO(int roundNumber, List<PlayedCardDTO> playedCards, long currentTurnUserId)

PlayCardDTO (Long gameId, Long playerId, String card, int roundNumber)

PlayedCardDTO (Long gameId, Long playerId, String card)

PlayerDTO (Long userId, String username, Bool isAIPlayer)

### Leaderboard Management

LeaderBDTO (List <UserDTO>)

LeaderBEntryDTO (Long userId, int points)

LeaderBUpdateDTO (Long gameId, String token, List <LeaderBEntryDTO> updates)

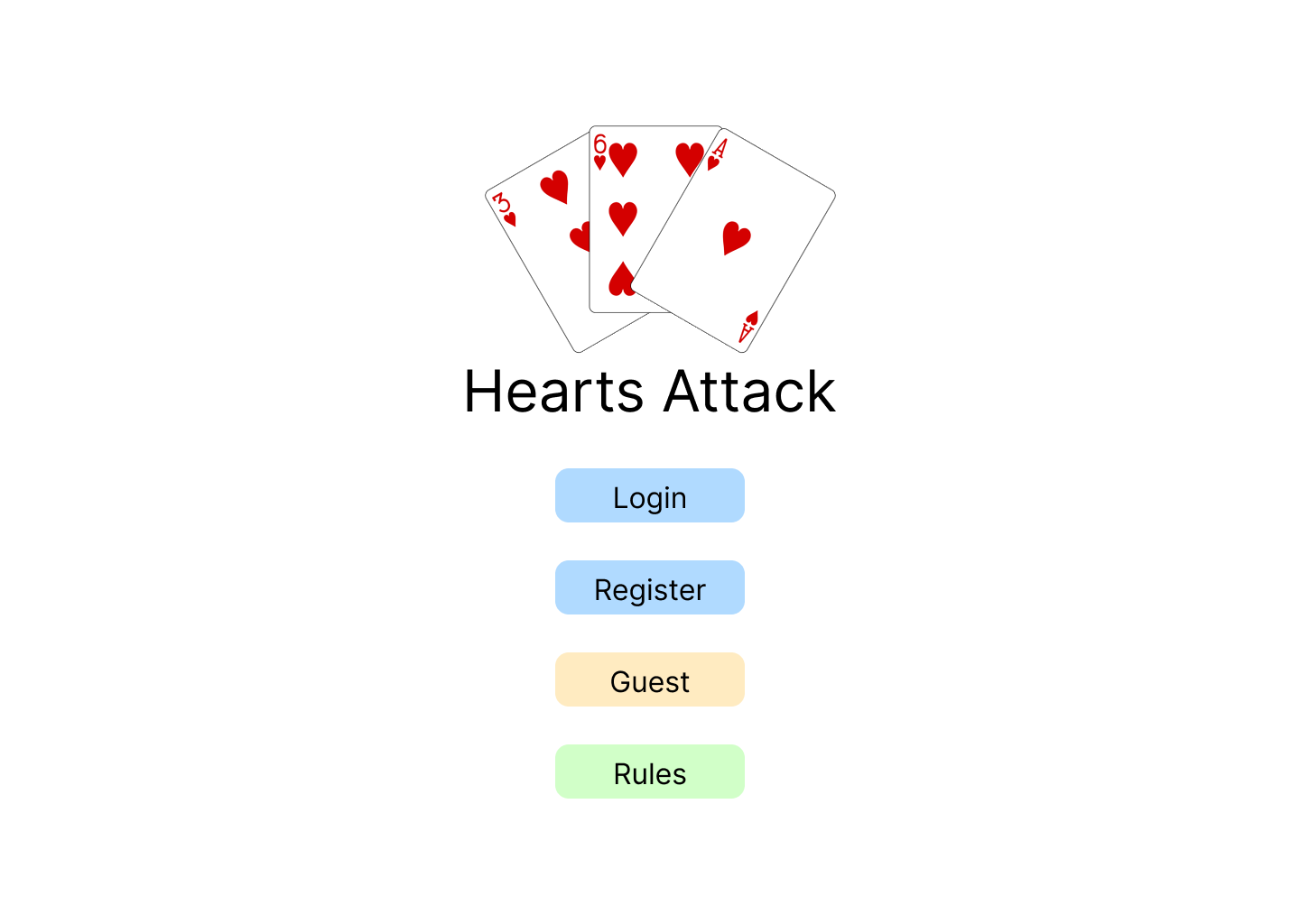
### External API

The external API we will rely on is: “Deck of Cards: an API <https://deckofcardsapi.com/>”

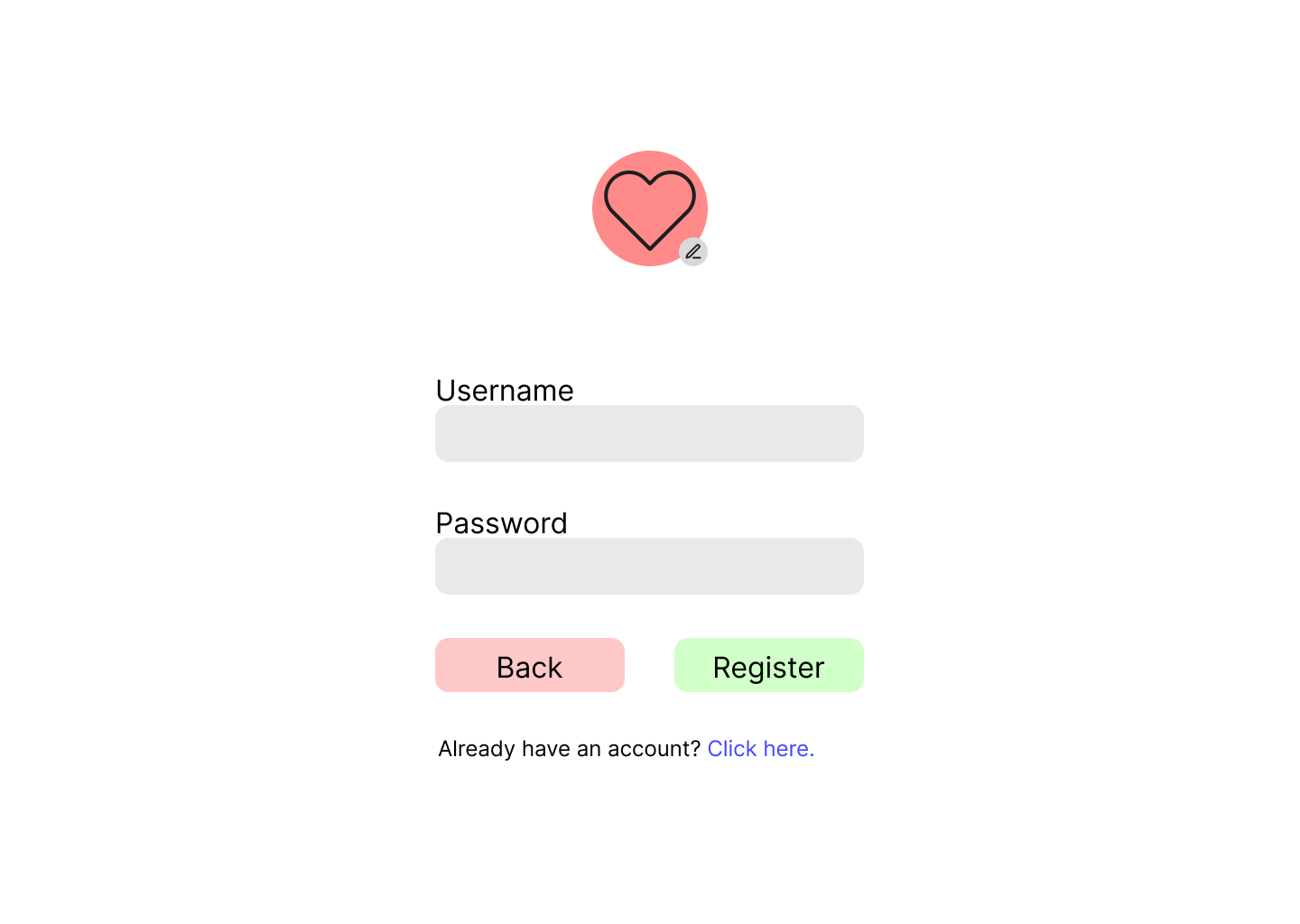
In Requests sent to that API, individual cards have names consisting of two alphanumeric symbols with the first representing the value as follows: [A] for ace, [2-9] for that exact card value, [0] for 10, [J] for jack, [Q] for queen and [K] for king. The second symbol denotes the suit as follows: [S] for Spades (♠), [D] for Diamonds (♦), [C] for Clubs (♣) and [H] for Hearts (♥). We will use the same nomenclature: So a legal card name is: /[02-9AJKQ][CDHS]/.

# Mockup

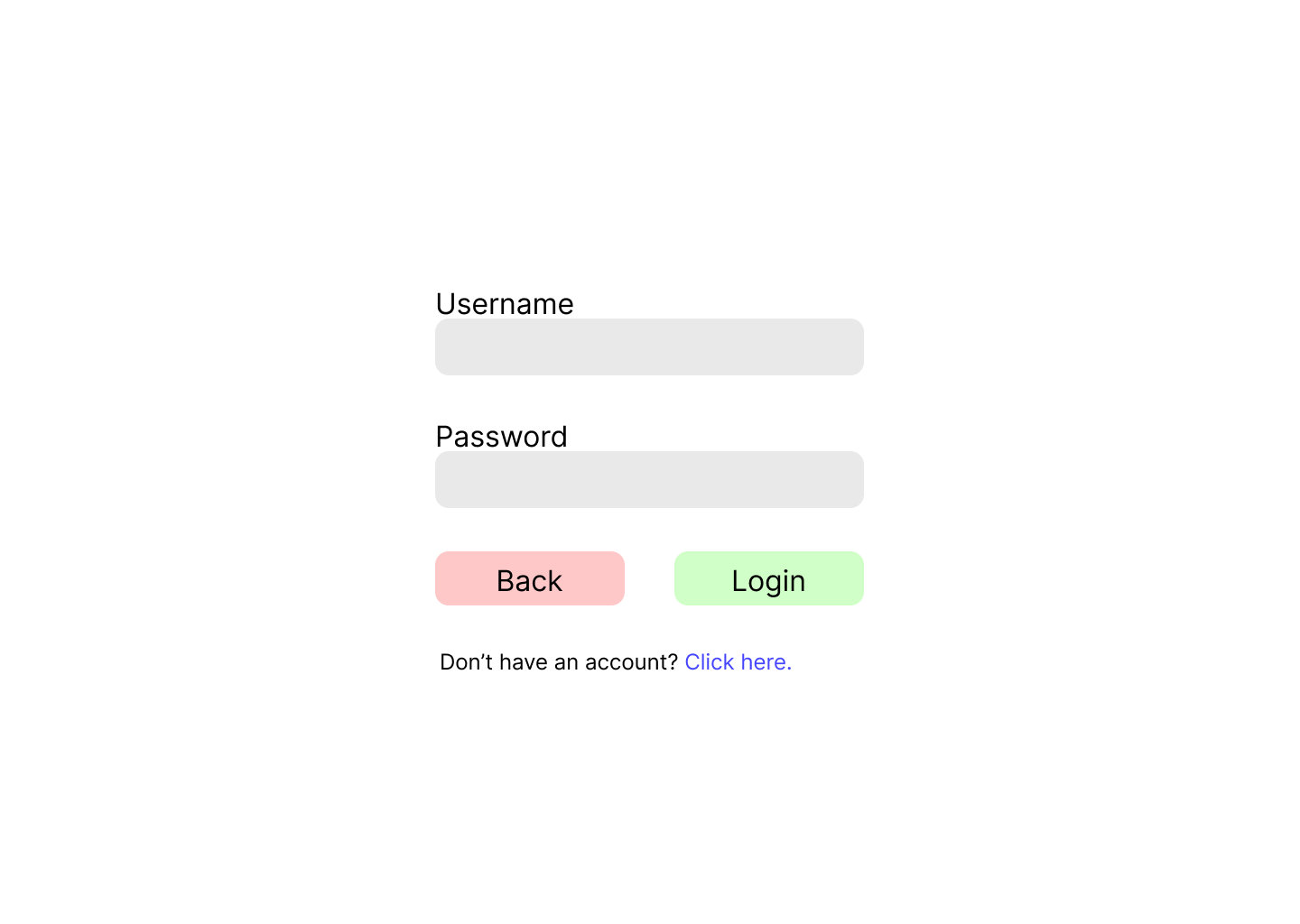
**Landing Page:**

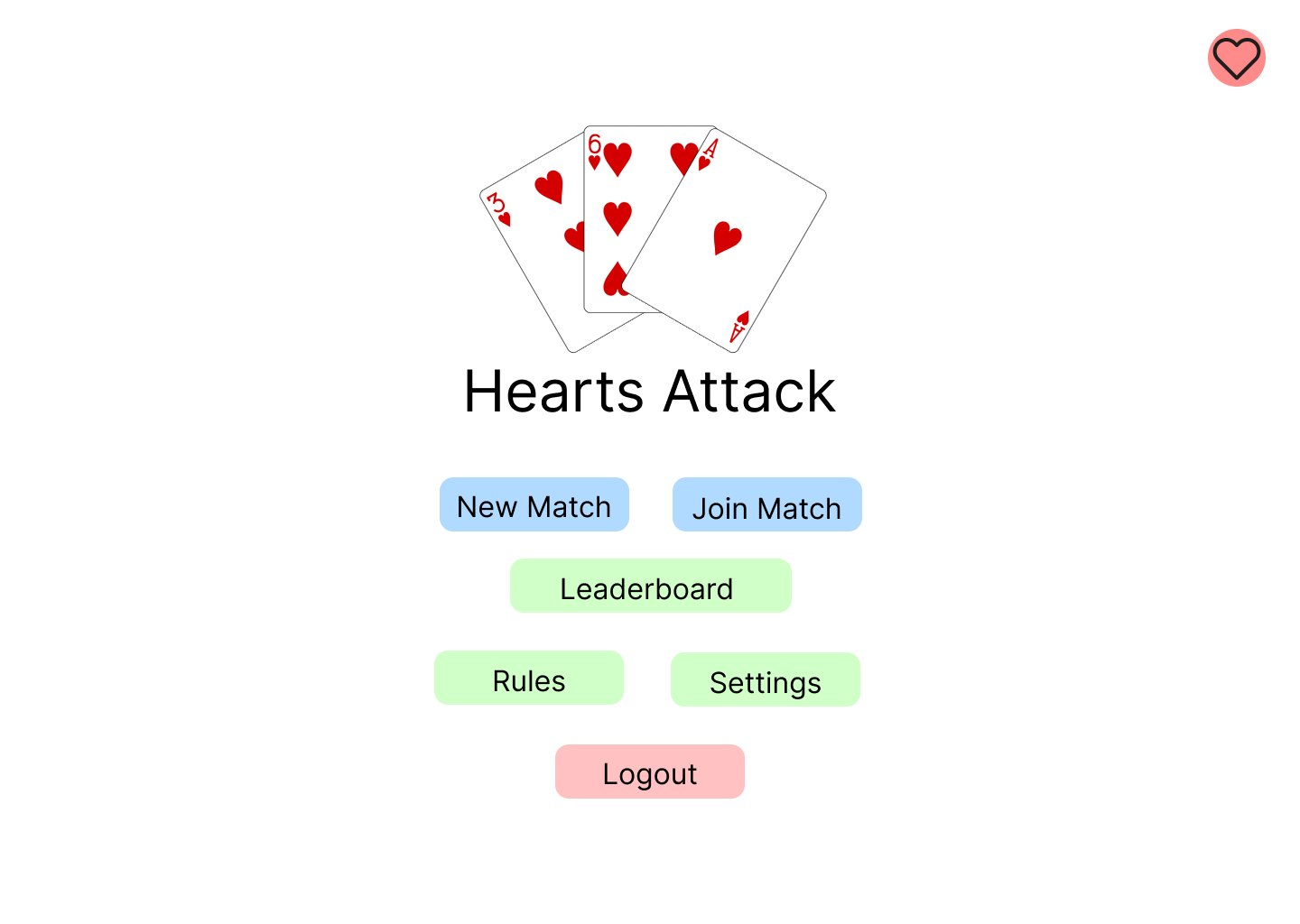


**Register Page:**

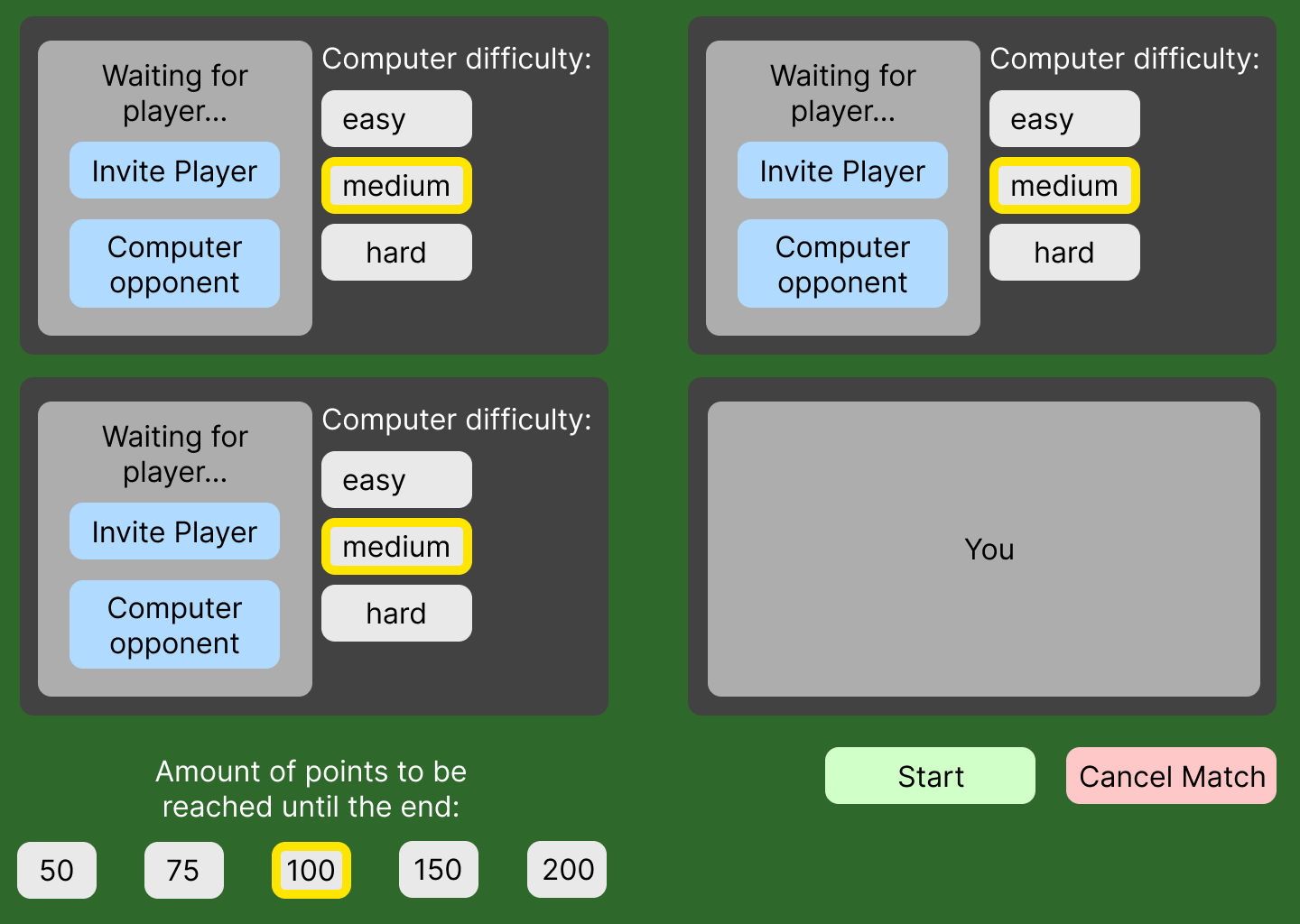


**Login Page:**

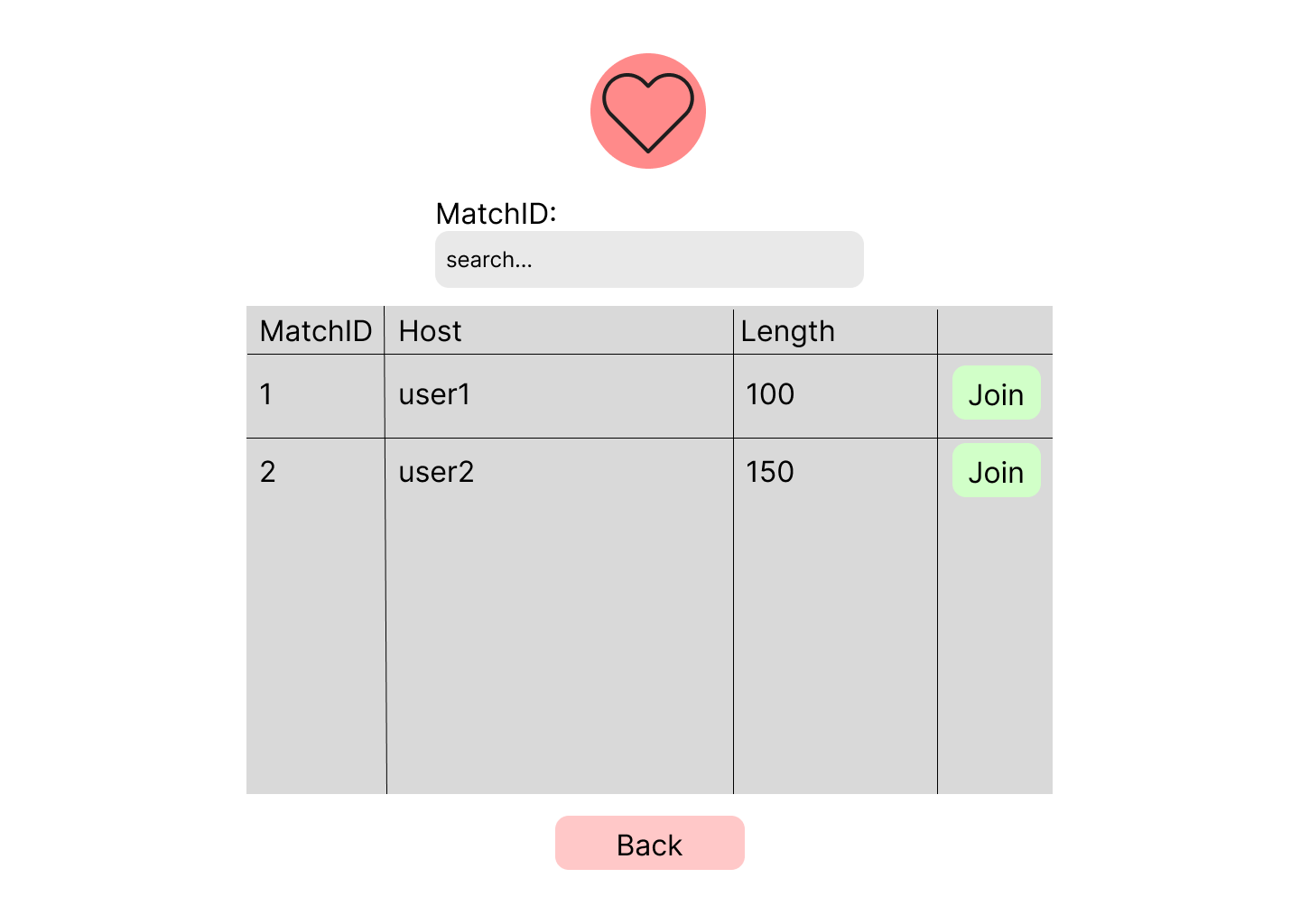
**Landing Page User:**



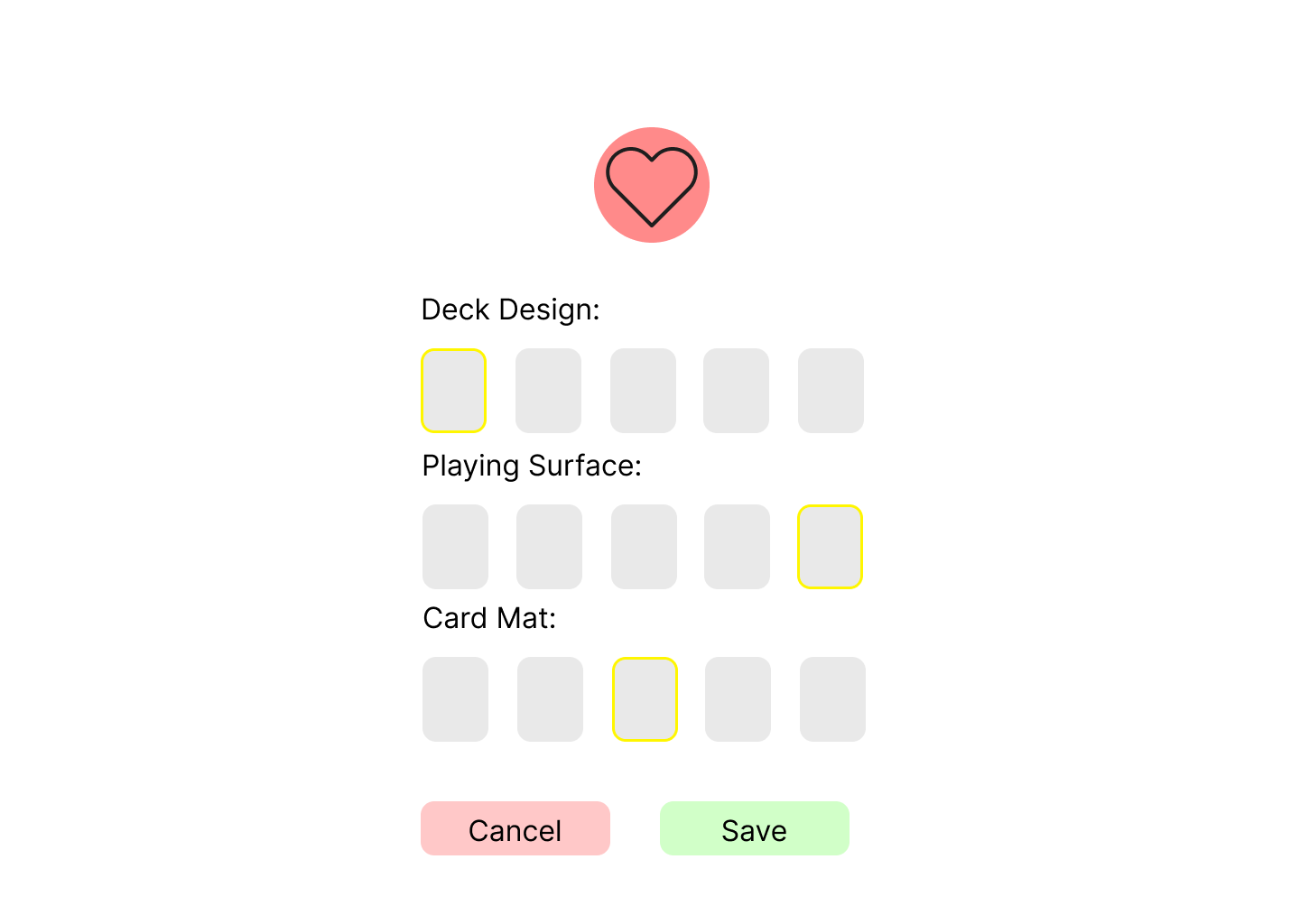
**Start Page:**



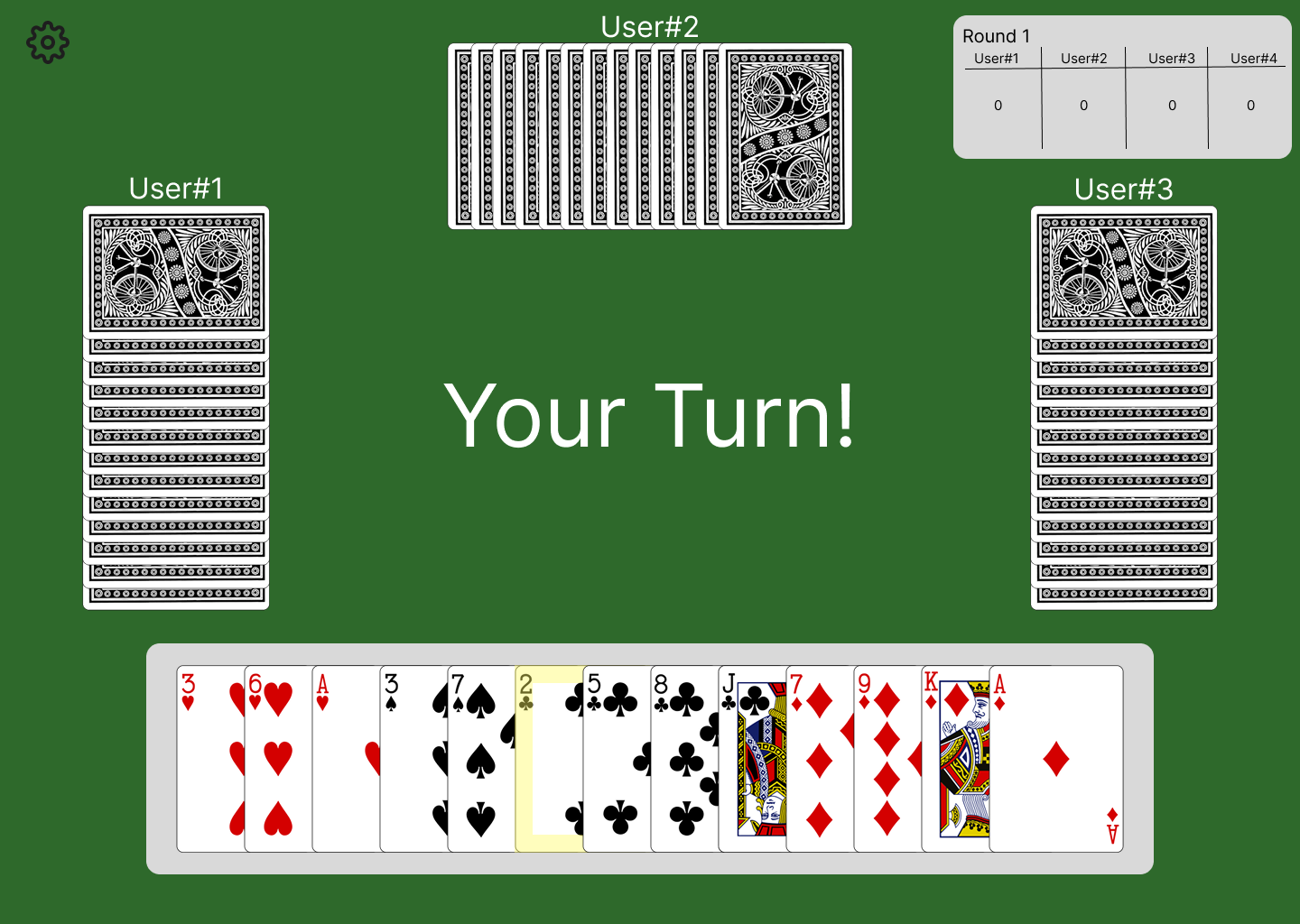
**Join Page:**



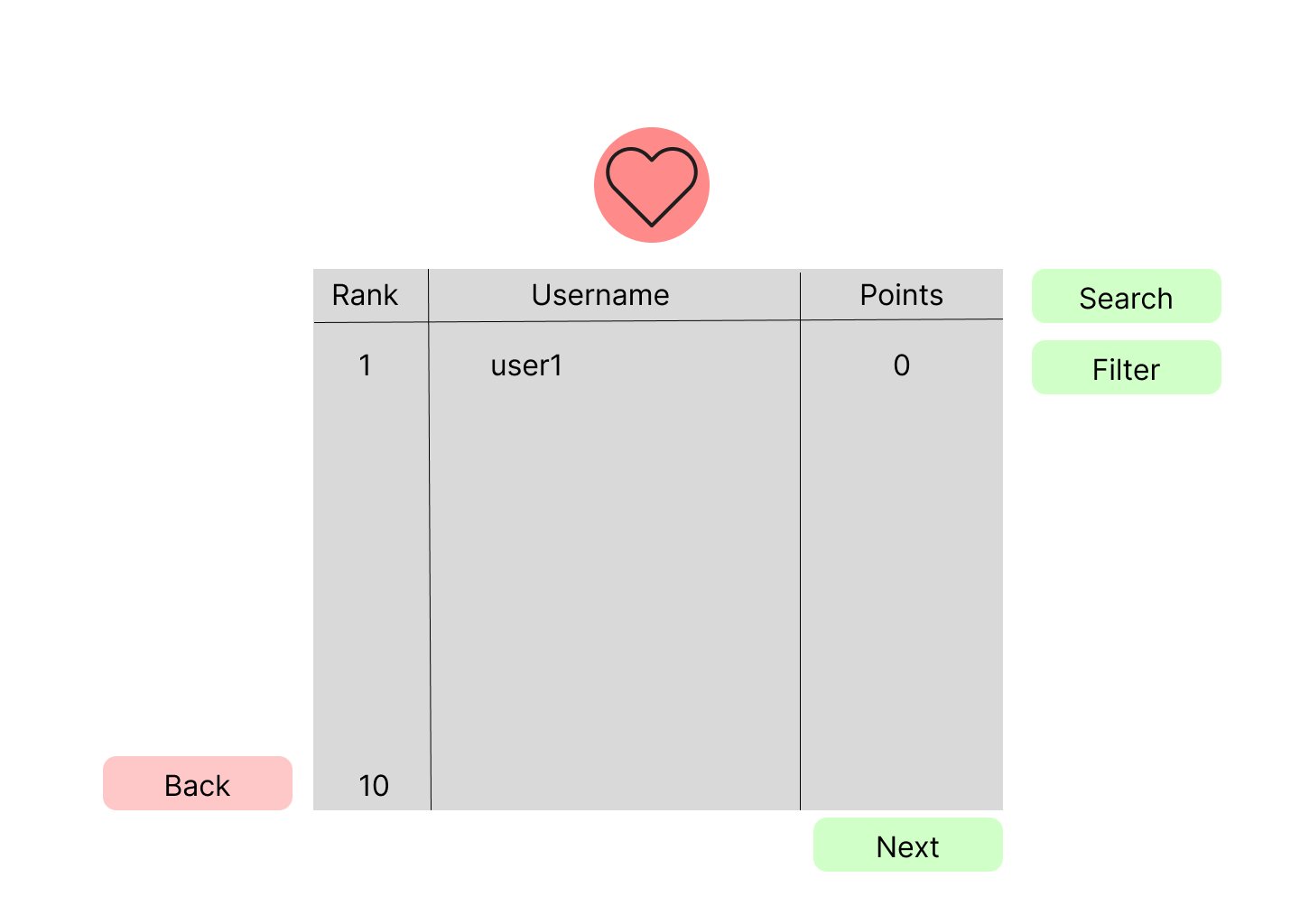
**Settings Page:**

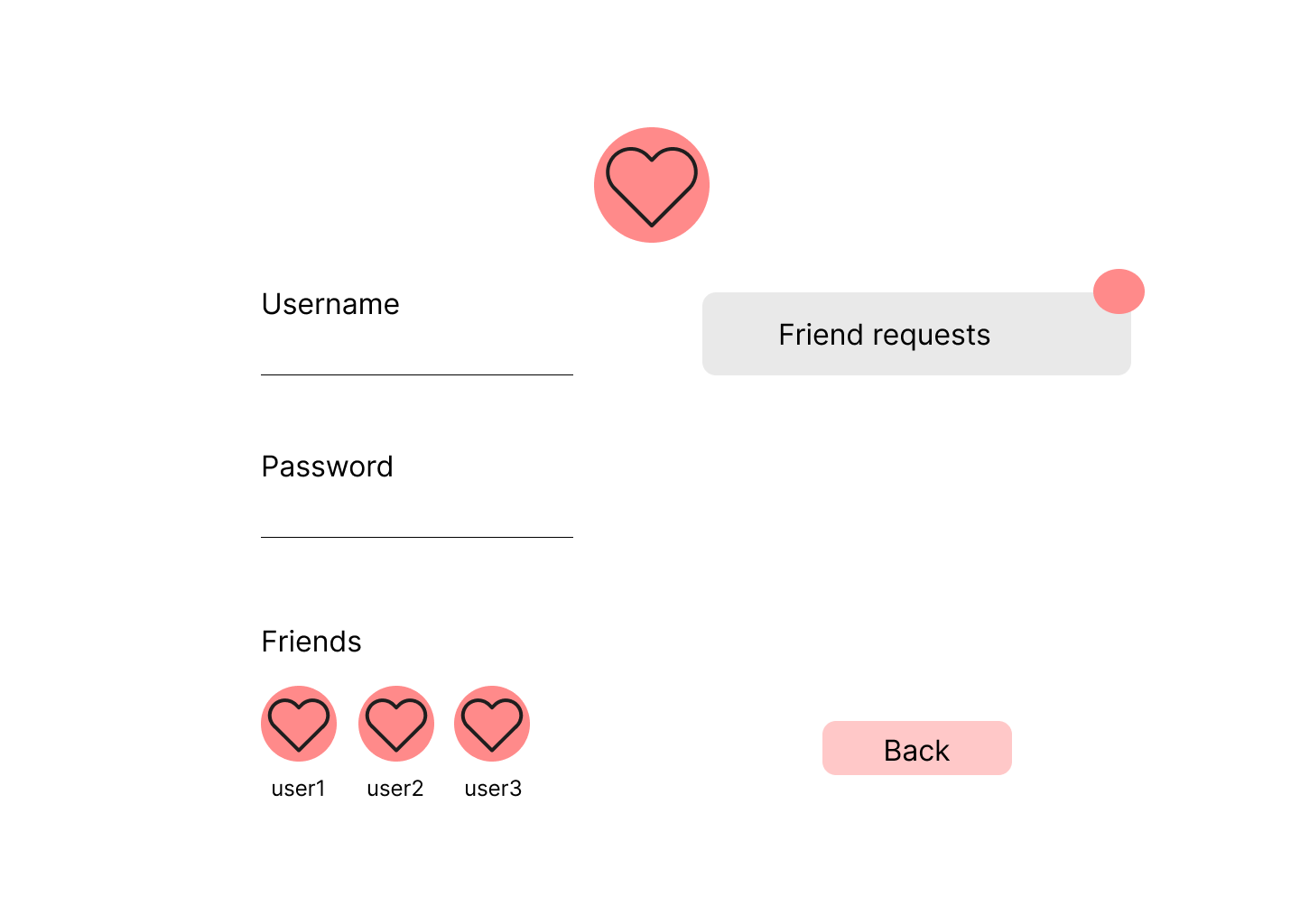


**Match Page:**



**Leaderboard Page:**

**Profile Page (own):**



**Profile Page (others):**

