**RELAZIONE PROGETTO PER IL CORSO “TECNOLOGIE INTERNET”**

**MARCO BAREZZI (matr. 311349) – MARCO CALVI (matr. 309307)**

**Client Peer-to-Peer per il gioco degli scacchi**

**INTRODUZIONE:**

Il progetto realizzato consiste in un client per giocare a scacchi attraverso una connessione Peer-to-Peer, che si interfaccia a un database MySQL per il salvataggio degli utenti e delle partite tra di essi.

**STRUTTURAZIONE DEL PROGETTO E LIBRERIE ESTERNE:**

Il progetto è realizzato utilizzando il framework JavaScript “React”, che permetta la creazione di interfacce utente altamente efficienti, caratterizzate da uno “stato” che permette di renderizzare nuovamente solo i componenti che hanno subito delle modifiche.

Il progetto è diviso in frontend e backend.

Nel frontend sono presenti le UI che vengono mostrate agli utenti, la logica dietro il loro alternarsi e le meccaniche di gioco, e sempre nel frontend viene inizializzata la connessione tra i due Peer che intendono giocare una partita.

Nel backend invece, vengono gestite le richieste in arrivo dai Client attraverso un meccanismo request-response, e avvengono le interazioni con il server MySQL.

**Le librerie utilizzate per il frontend sono:**

* **PeerJS** – Permette la creazione di Peer identificati unicamente attraverso un ID, tra i quali viene instanziata una connessione end-to-end.
* **React** – La principale libreria utilizzata per la creazione delle UI.
* **React-Router-Dom** – Responsabile dell’alternarsi delle UI.
* Librerie grafiche aggiuntive (**ReactIcons**, **React-Bootstrap**, **Material-UI**) – Per lo styling di componenti custom e l’utilizzo di altri già creati.

**Le librerie per il backend sono:**

* **ExpressJS** – Framework per applicazioni web Node.js flessibile e leggero che fornisce una serie di funzioni avanzate per le applicazioni web, e con cui viene istanziato il server a cui il Client effettuerà le richieste.
* **Sequelize** – Per l’interazione con MySQL attraverso API compatibili.

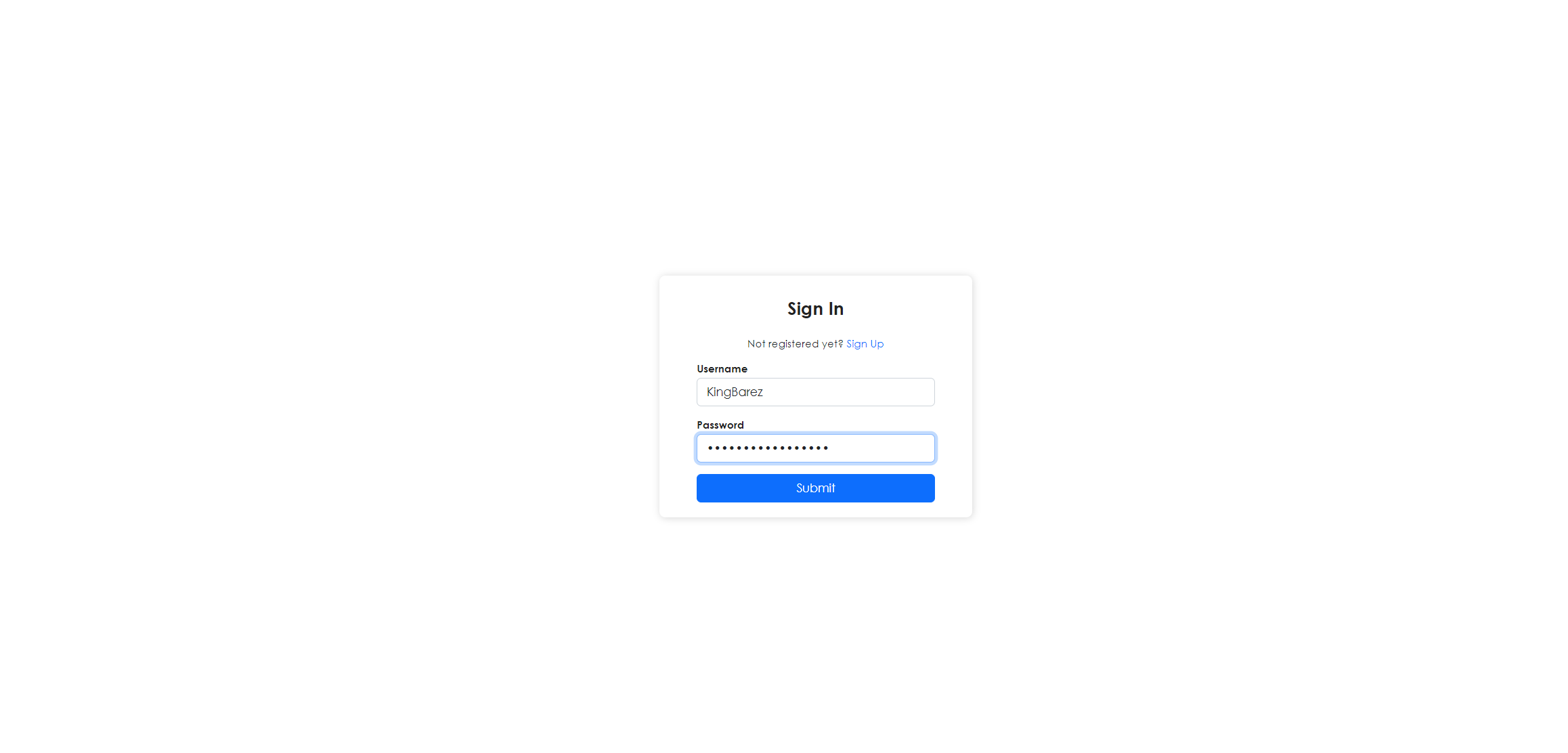
Tra frontend e backend, per la realizzazione di richieste HTTP e la ricezione delle risposte, viene utilizzato Axios, che attraverso il meccanismo delle Promise permette la comunicazione asincrona tra i due fronti.

Oltre alle librerie sopra descritte, sono stati utilizzati due tools particolarmente utili:

* **Xampp:** Per la creazione del server MySQL e la sua facile gestione.
* **Nodemon:** Per il riavvio automatico del server Express alla rilevazione di cambiamenti nel backend.

**FUNZIONALITA’ E UTILIZZO:**

All’avvio del client, viene mostrata la pagina di login attraverso cui gli utenti possono autenticarsi attraverso l’inserimento delle credenziali, che verranno confrontate con quelle presenti sul server MySQL.



*Pagina di login*

Se l’esito del tentativo di login è positivo, all’utente viene mostrata la pagina principale del Client, in cui sono presenti informazioni sul client stesso e le statistiche personali dell’utente, mentre in caso di esito negativo l’utente viene invitato a inserire nuovamente le credenziali, dopo la comunicazione dell’errore riscontrato.

Immagine che contiene testo

Descrizione generata automaticamente

*Homepage del client*

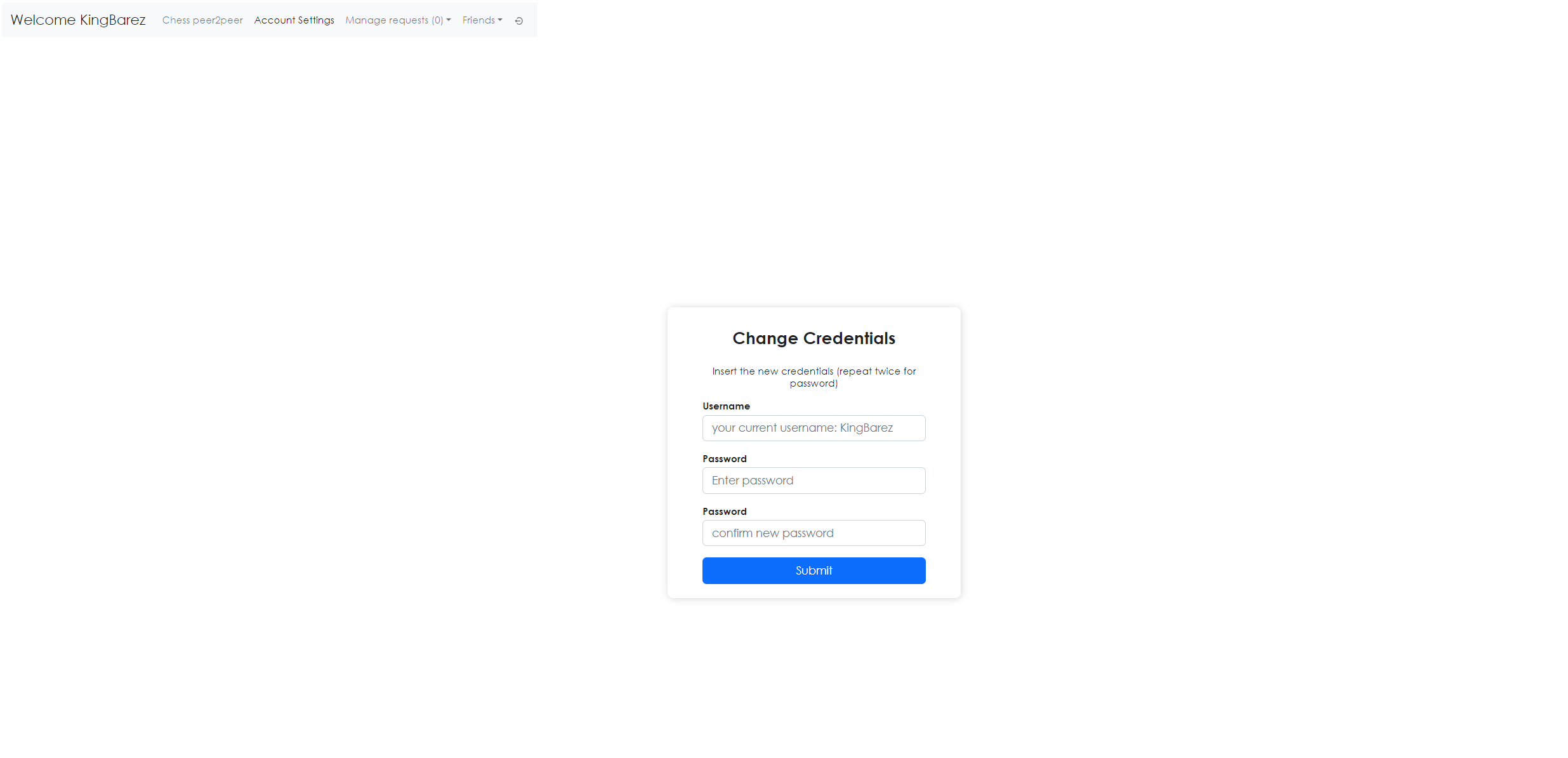
Sempre nella pagina di login è presente un tasto che permette di registrarsi come nuovo utente, attraverso un form di registrazione che sempre in seguito a controlli sul database (unicità dell’username), garantisce l’iscrizione come nuovo utente.

Tutte le UI mostrate da autenticati, sono caratterizzate da una barra di navigazione che permette la transizione da una all’altra, oltre che di sfidare un amico tra quelli online e di visualizzare le richieste di amicizia e di sfida attraverso dei menù a tendina.

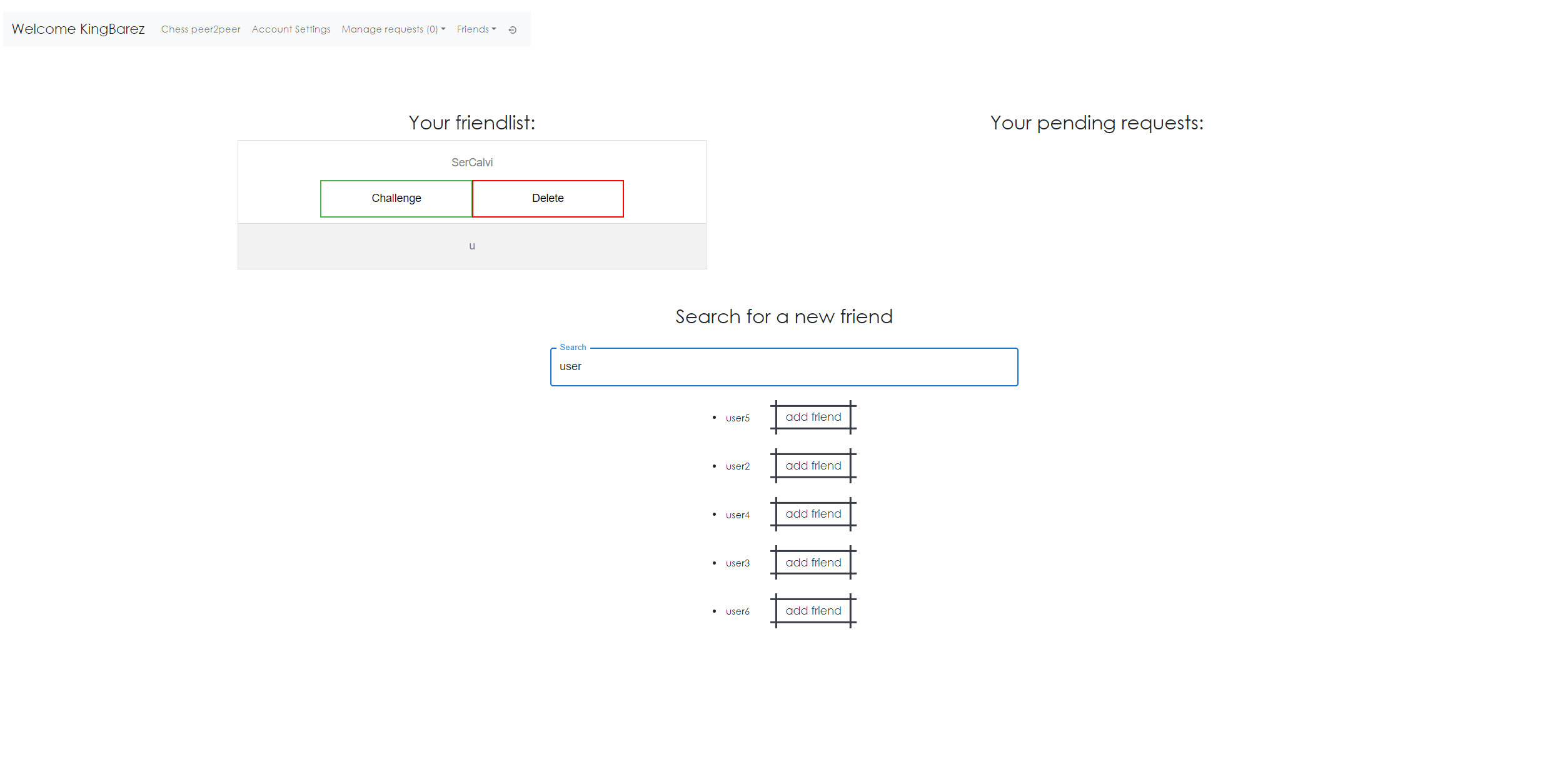


*Barra di navigazione delle pagine da autenticato*

Le pagine accessibili dalla barra di navigazione sono quella relativa al cambio di credenziali, la home, quella relativa alla gestione delle amicizie, da cui è possibile sfidare un amico o effettuare una richiesta di amicizia tramite ricerca di username tra quelli presenti nel database, oltre che quella che riporta alla pagina di login attraverso il logout della propria utenza.



*Pagina per il cambio delle credenziali*

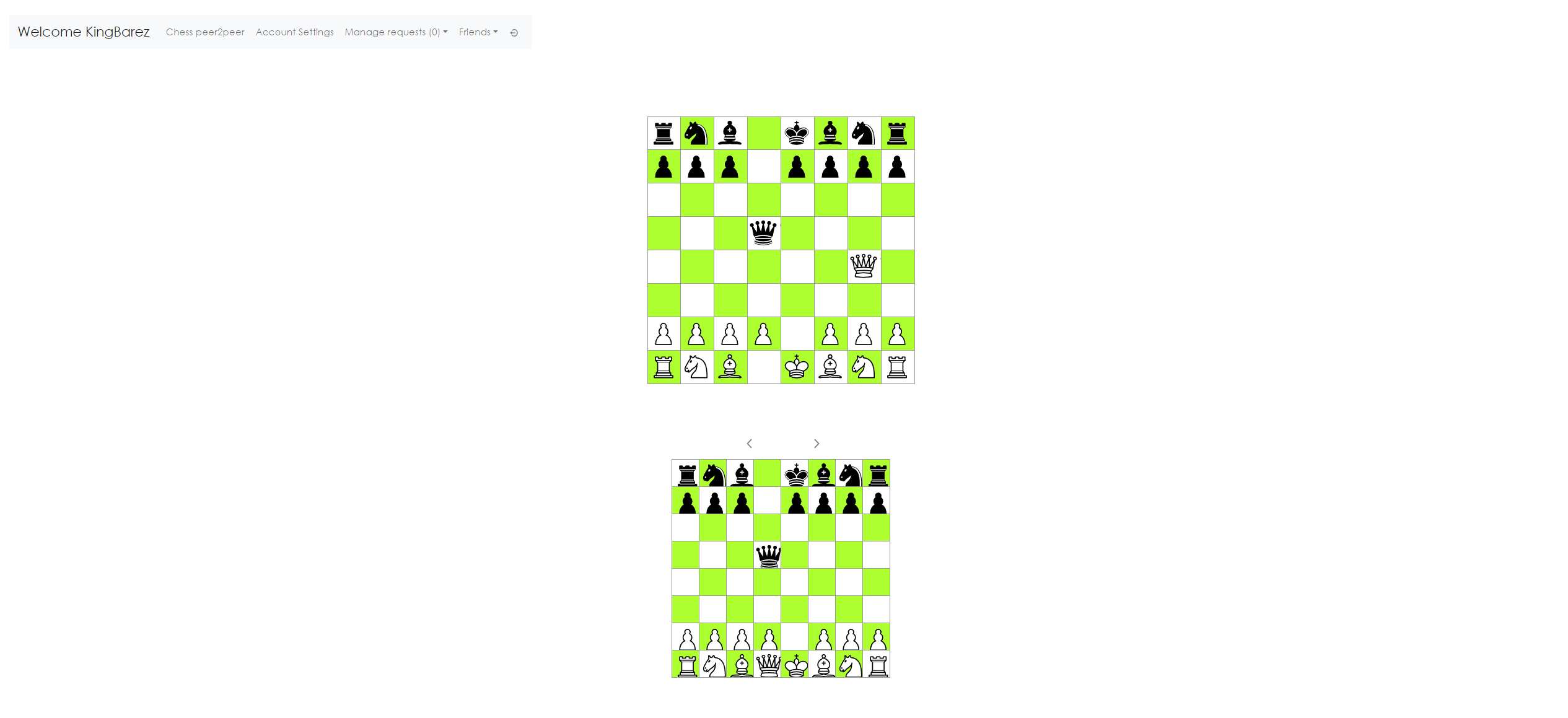


*Pagina di gestione delle amicizie*

Una volta sfidato un proprio amico per una partita, lo sfidante viene rediretto alla schermata di gioco, in cui attende che l’amico accetti la partita, e viene subito creato il Peer, come combinazione di una stringa pre-determinata, dell’username (univoco) dell’utente, e dell’id della partita creata.

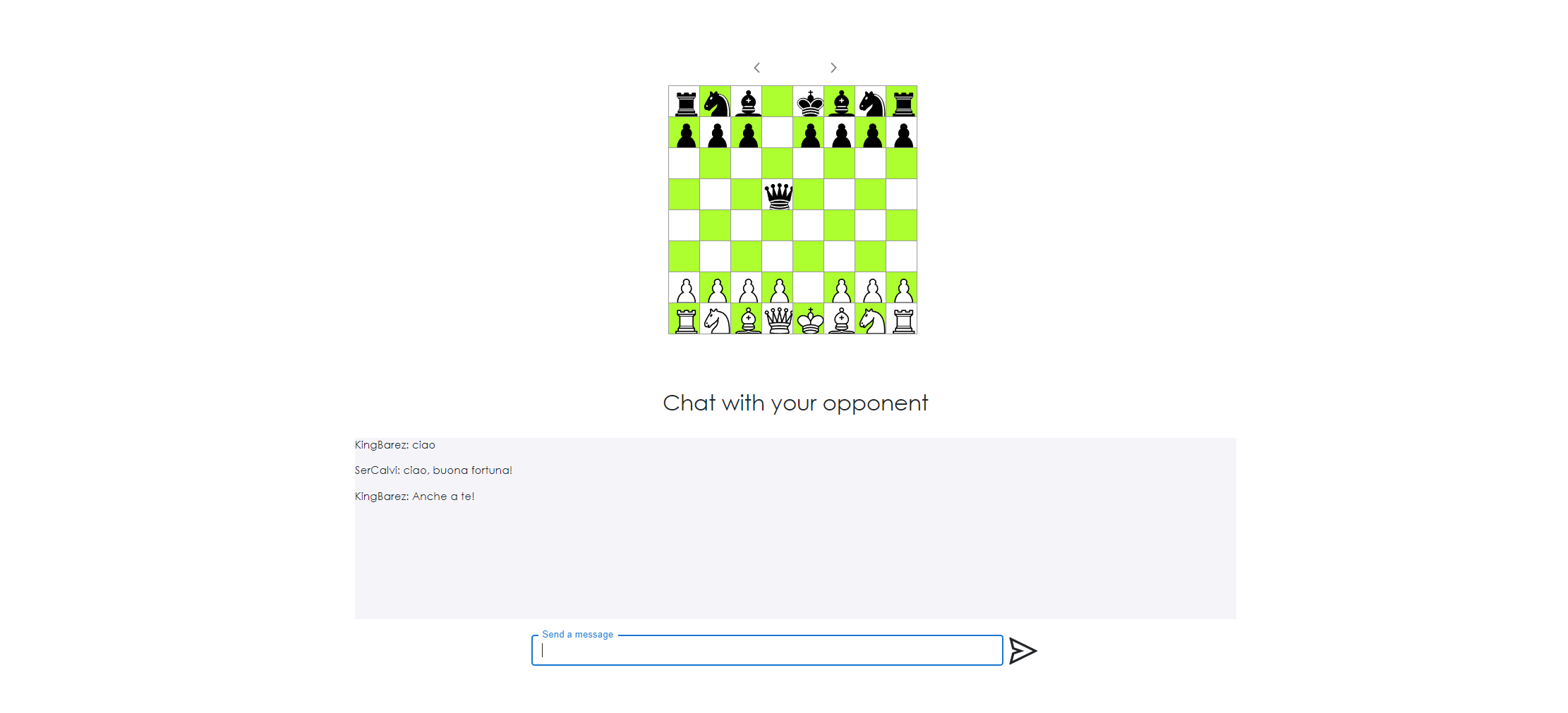
Nel caso in cui l’amico sfidato accetti la partita, anche a quest’ultimo viene mostrata l’interfaccia di gioco e viene creata l’istanza anche del suo Peer, mentre lo sfidante viene avvisato dell’accettazione della partita tramite un alert di sistema, e viene inizializzata la connessione end-to-end.

A questo punto lo sfidante (che gioca con il bianco) può effettuare la prima mossa, e la partita comincia.



*Particolare della schermata di gioco – scacchiera di gioco e di history*

La schermata di gioco, presenta la scacchiera di gioco, su cui vengono effettuate le mosse (vengono mostrate come possibili solo quelle realmente effettuabili, considerando anche gli eventuali scacchi), una seconda scacchiera più piccola, su cui è possibile verificare gli stati precedenti della partita muovendosi con delle frecce, e la chat, tramite cui è possibile lo scambio di messaggi tra i due giocatori.



*Particolare della schermata di gioco – scacchiera di history e chat*

Le mosse vengono passate inviando sulla connessione aperta lo stato aggiornato della scacchiera, che viene ridisegnata aggiornata nelle schermate di entrambi i giocatori, mentre i messaggi sono scambiati in formato plain-text e aggiunti a una lista che viene mappata su un contenitore in modo da mostrare lo storico dei messaggi.