

Iterazione 4

1.0 Introduzione

La quarta iterazione si concentrerà sui seguenti aspetti:

- Implementazione di un'interfaccia grafica interattiva che permetta agli organizzatori e ai giocatori di utilizzare l'applicazione in tutti i casi d'uso previsti.
- Piccole modifiche ai casi d'uso relative alle estensioni, sia a livello di codice che di diagramma UML implementati precedentemente.
- Refactoring di tutto il codice dell'applicazione per uniformarsi alla GUI e alle ultime versioni dei casi d'uso

1.1 Aggiornamento dei casi d'uso

Come per le precedenti iterazioni, durante questa fase sono stati specificati e rivisti dei dettagli dei casi d'uso implementati, in particolare per quanto riguarda le estensioni, per essere in accordo con l'effettivo funzionamento dell'applicazione. Per visionare le modifiche apportate ai suddetti, si consulti il documento “Modello dei casi d'uso” riportato nella cartella in versione aggiornata per questa iterazione.

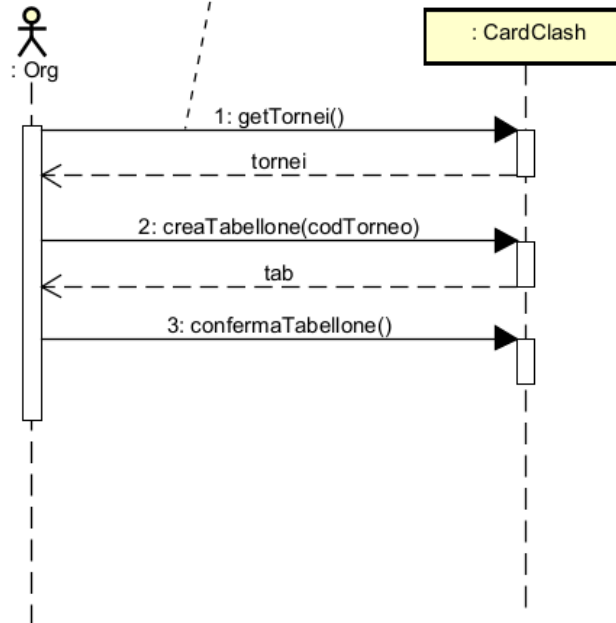
2.0 Aggiornamento diagrammi di sequenza di sistema

Nella fase di analisi della quarta iterazione, sono state modificati alcuni SSD e SD in vista dell'implementazione della GUI, di seguito le modifiche:

- **UC5:** Creazione tabellone

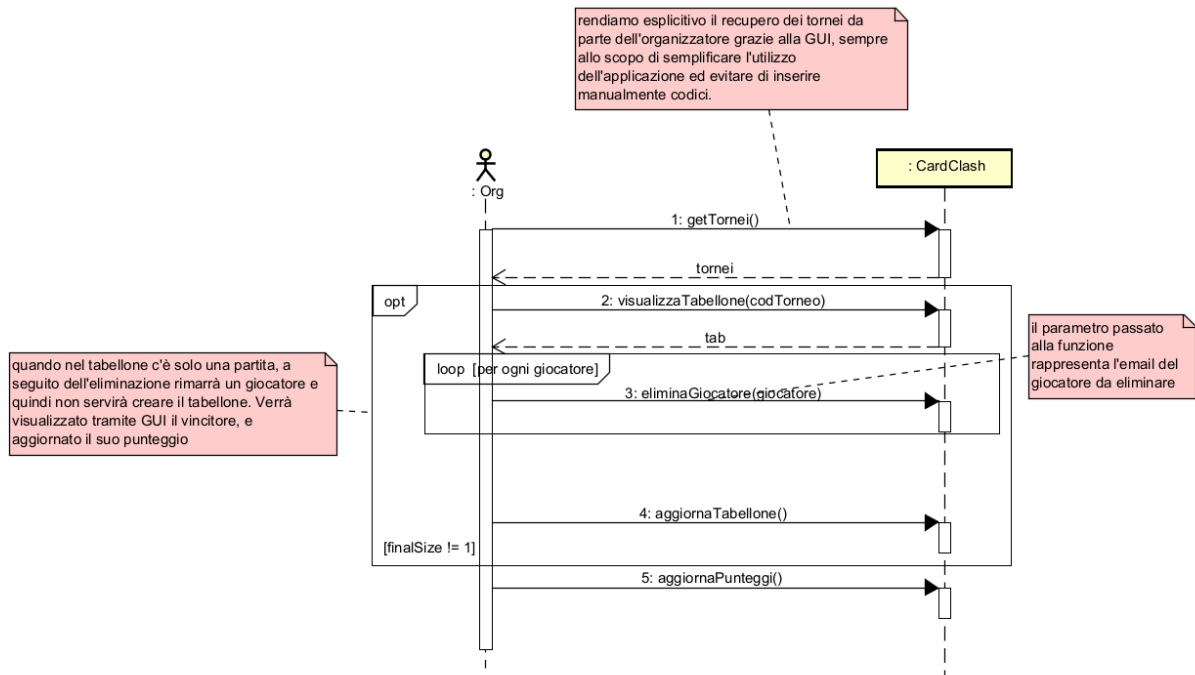
In questo SSD è stato aggiunto un getter per semplificare l'interazione tra il sistema e la GUI. In particolare, i tornei con data già trascorsa verranno filtrati, mentre quelli selezionabili saranno mostrati a schermo tramite un menu a tendina, evitando così l'inserimento manuale dei codici torneo.

classico getter aggiunto per semplificare l'interazione col sistema tramite GUI ed evitare di inserire codici manualmente. Inoltre, saranno filtrati i tornei con data passata, per cui risulta inutile creargli un tabellone.



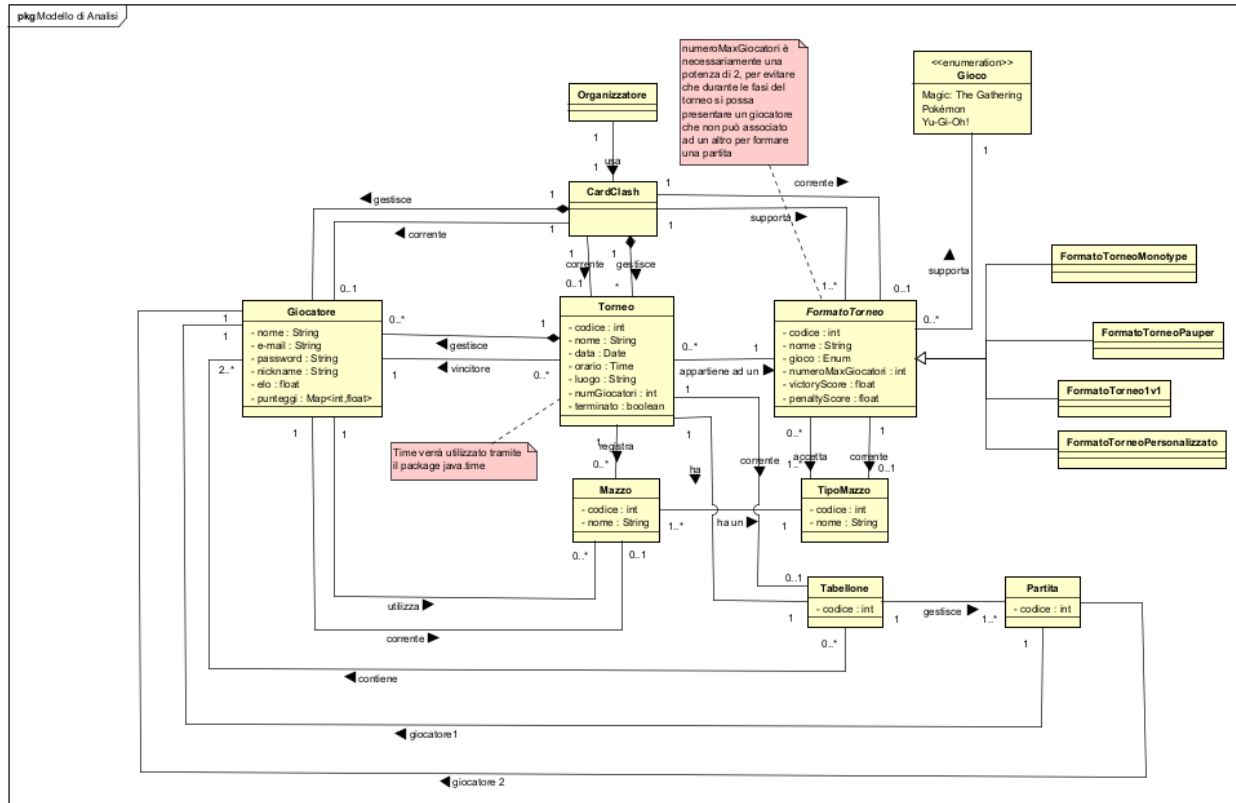
- **UC6:** Gestisci eliminazione

Anche in questo SSD è stato aggiunto un getter. Inoltre, è stato implementato un controllo tramite Guard per consentire l'aggiornamento dei tabelloni e l'eliminazione dei giocatori solo se la dimensione finale della lista dei giocatori nel tabellone è diversa da 1. In caso contrario, l'ultimo giocatore rimasto verrà dichiarato vincitore tramite un popup.



2.1 Modello di dominio attuale

Nonostante non sia stato modificato, riportiamo il modello di dominio dell'iterazione corrente:

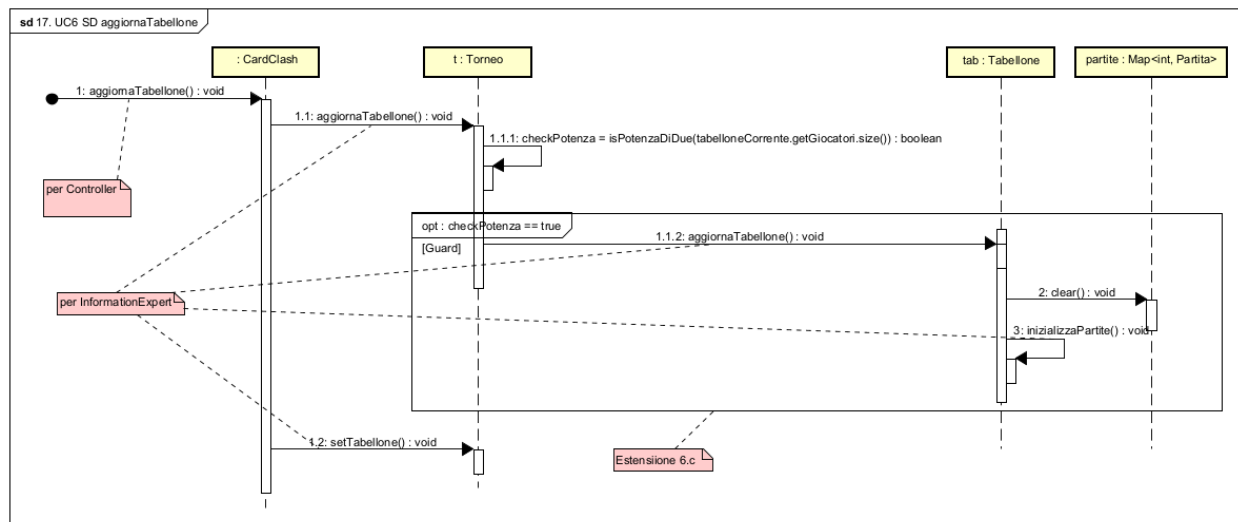


3.0 Fase di Progettazione

Per quanto riguarda la fase di progettazione, sono state effettuate alcune modifiche circa i diagrammi di sequenza, in modo tale da implementare numerose estensioni; di seguito le modifiche:

3.1 Modifica SD aggiornaTabellone(UC6)

Il seguente diagramma è stato aggiornato per controllare che la creazione del nuovo tabellone rispetti sempre la condizione che il numero dei giocatori sia potenza di due, questo per evitare che inavvertitamente si possa eliminare un numero di giocatori inadeguato per l'aggiornamento del tabellone, in accordo con l'estensione 6c.



3.2 Diagramma delle classi corrente

Riportiamo per completezza l'attuale diagramma delle classi del sistema:

4.0 Estensioni implementate tramite codice

Numerose estensioni sono state implementate lato codice, e ciò non ha previsto la necessità di aggiornare i diagrammi UML. Le riportiamo di seguito:

- **1b:** la GUI mostra un messaggio di errore se l'amministratore vuole creare un torneo in una data in cui è già previsto un altro.
- **3b:** è stato modificato il funzionamento del metodo isAperto(), utilizzata dall'SD mostraTorneiDisponibili, in particolare dalla lista dei tornei a cui è possibile iscriversi sono rimossi i tornei a cui si è già iscritti o che hanno già un tabellone (e che quindi sono già in corso). Insieme ad esso, è stato raffinato il testing di tale metodo, si consulti il codice per vedere le migliorie.
- **4a e 4b:** queste due estensioni sono semplici controlli del corretto inserimento dei dati, per verificare che la tipologia non sia già esistente.
- **6b:** la GUI mostra un semplice messaggio di errore se il giocatore inserito non è presente nel tabellone.
- **6d:** la GUI mostra un popup di alert, se eliminata l'ultima partita e rimasto l'ultimo giocatore non sia più possibile creare ulteriori tabelloni e che quindi questo sia il vincitore; si può dunque procedere all'aggiornamento dell'ELO.
- **8a e 8b:** estensioni di controllo implementate tramite GUI, per non permettere l'aggiornamento dell'ELO qualora il torneo non sia ancora iniziato o non sia ancora finito.

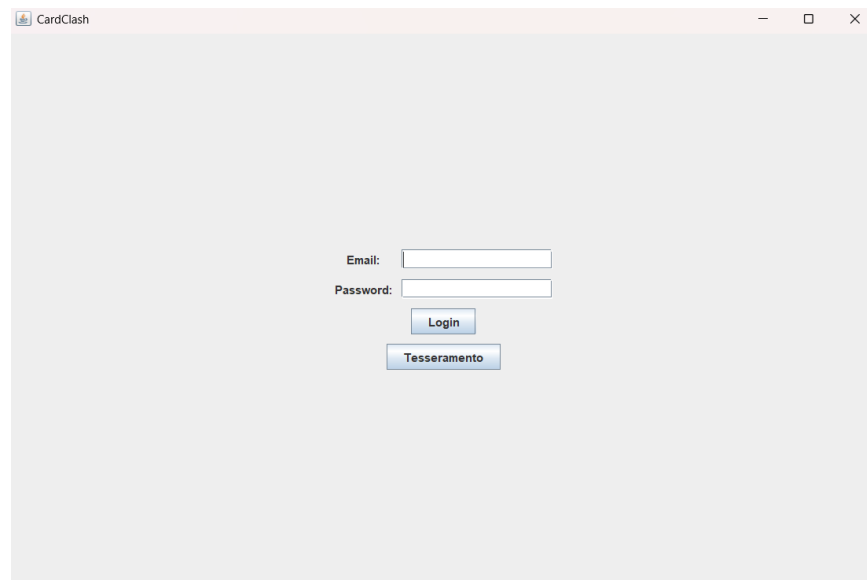
In aggiunta a ciò, sono state gestite numerose funzioni di rollback dei dati in caso di problematiche di utilizzo dell'applicazione a runtime.

Infine, l'organizzatore può liberamente eliminare dal torneo anche due giocatori che partecipavano alla stessa partita: questo potrebbe accadere nel caso di ritiri/squalifiche. In future iterazioni, l'estensione 6a (al momento, dunque, non totalmente implementata) si potrebbe modellare come caso d'uso/scenario alternativo separato, e tramite un nuovo elemento della GUI sottrarre punti ai giocatori, in accordo con le Regole di Dominio e del formato stesso.

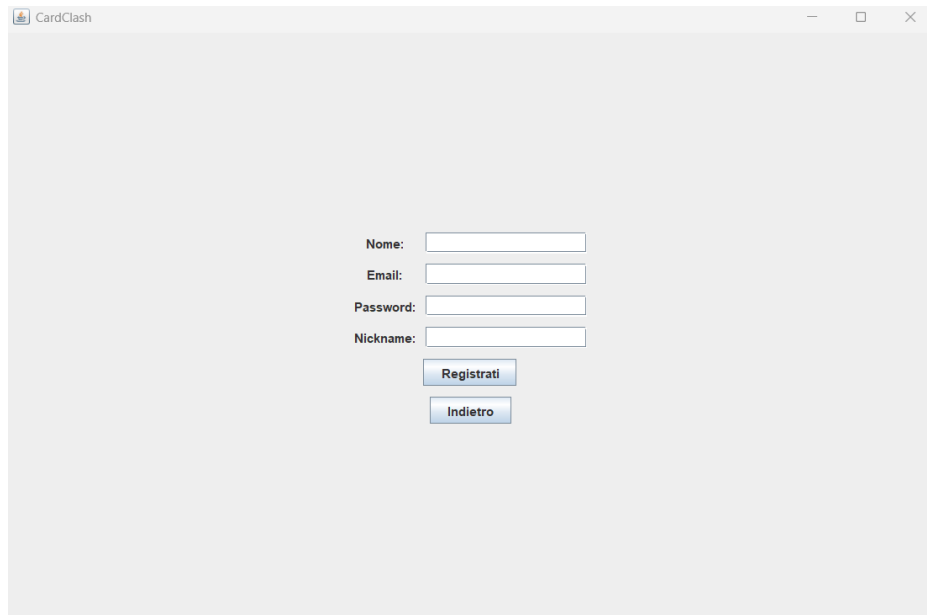
5.0 Interfaccia grafica (GUI)

In questa sezione riportiamo degli screenshot dell'interfaccia dell'applicazione in funzione.

5.1 Login



5.2 Tesseramento



A screenshot of a software window titled "CardClash". The window has a light gray background and standard window controls (minimize, maximize, close) in the top right corner. In the center, there is a registration form with four labels and corresponding text input fields: "Nome:", "Email:", "Password:", and "Nickname:". Below the input fields are two buttons: "Registrati" and "Indietro".

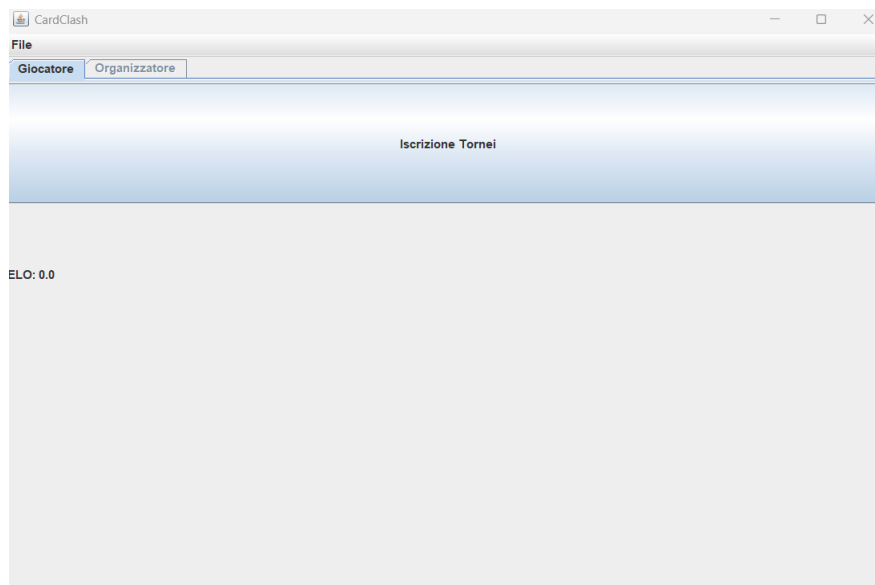
Nome:

Email:

Password:

Nickname:

5.3 Interfaccia lato giocatore



A screenshot of the "CardClash" application showing the player interface. The window has a title bar with "CardClash" and standard controls. Below the title bar is a menu bar with "File". Under the menu bar are two tabs: "Giocatore" (selected) and "Organizzatore". The main content area has a blue header bar with the text "Iscrizione Tornei". Below this, the text "ELO: 0.0" is displayed on the left side.

CardClash

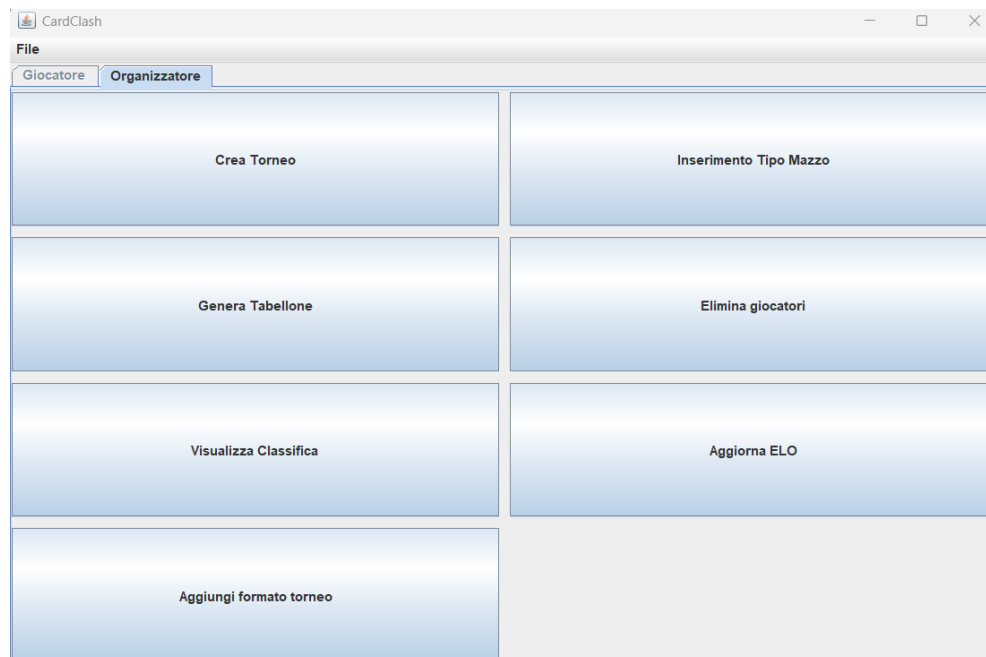
File

Giocatore Organizzatore

Iscrizione Tornei

ELO: 0.0

5.4 Interfaccia lato organizzatore



5.5 Form di creazione di un torneo

The screenshot shows the 'Form di creazione di un torneo' (Tournament Creation Form) in the CardClash application. The window has a menu bar with 'File' and a tab bar with 'Giocatore' and 'Organizzatore'. The main area contains a form with four input fields: 'Nome Torneo:', 'Data (yyyy-MM-dd):', 'Orario (HH:mm):', and 'Luogo:'. Below the fields are two buttons: 'Crea Torneo' and 'Indietro'.

5.6 Tabellone di un torneo

Tabellone Generato

Tabellone (non confermato)

codPartita	Giocatore1	Punti G1	Giocatore2	Punti G2
227734	ciao2@gmail.c...	0.0	ciao1@gmail.c...	0.0
557810	ciao@gmail.co...	0.0	ricky@gmail.co...	0.0

OK

5.7 Tabellone aggiornato dopo un'eliminazione

Tabellone Iniziale


Tabellone (789077)

codPartita	Giocatore1	Punti G1	Giocatore2	Punti G2
441339	ciao1@gmail.c...	2.0	ciao@gmail.co...	2.0

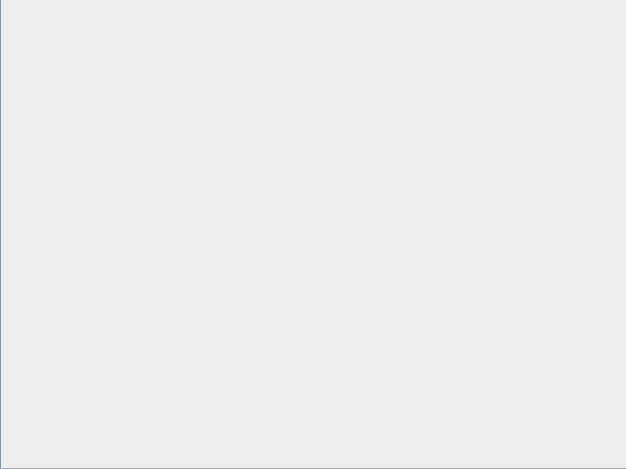
OK

5.8 Classifica

Classifica

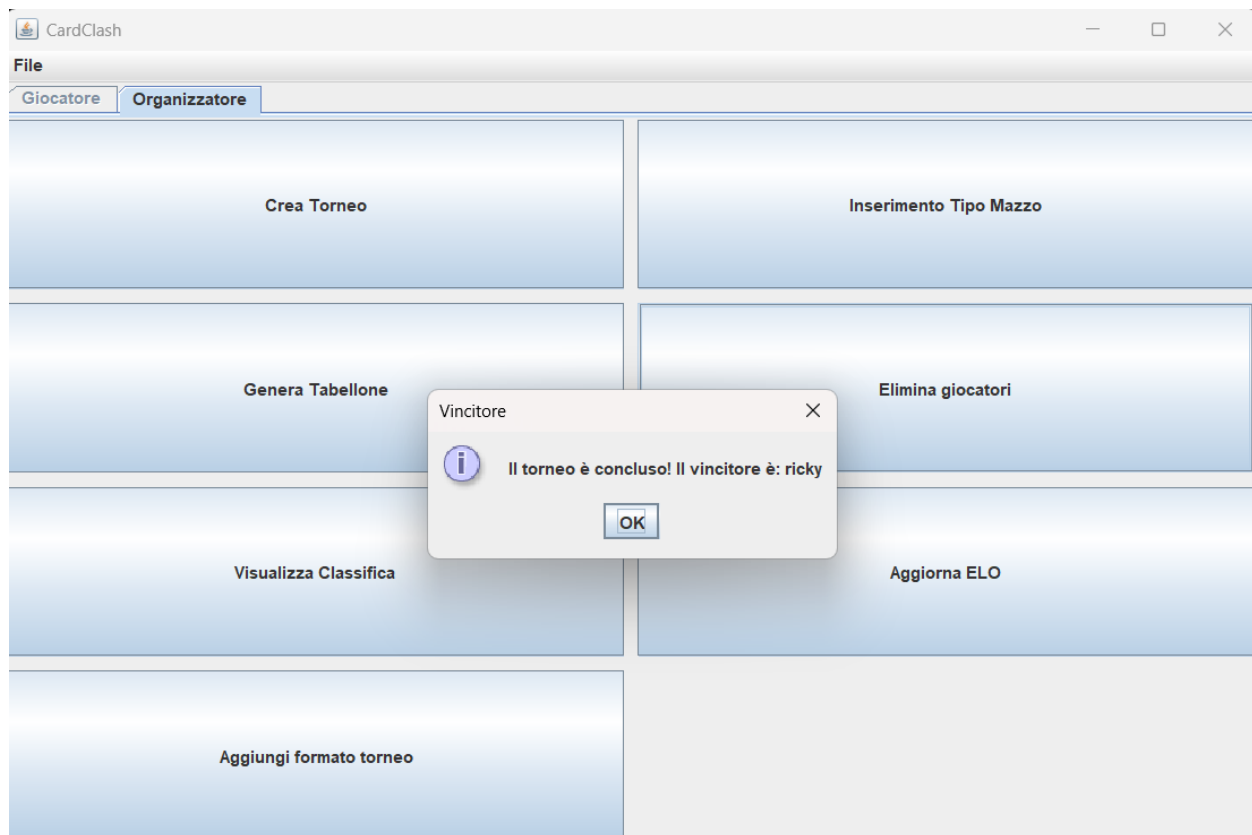


Posizione	Email	Nickname	Punteggio
1	ciao1@gmail.com	asfasf	2.0
2	ciao@gmail.com	ASFASF	2.0
3	ciao2@gmail.com	asfasf	0.0
4	ricky@gmail.com	ricky	0.0

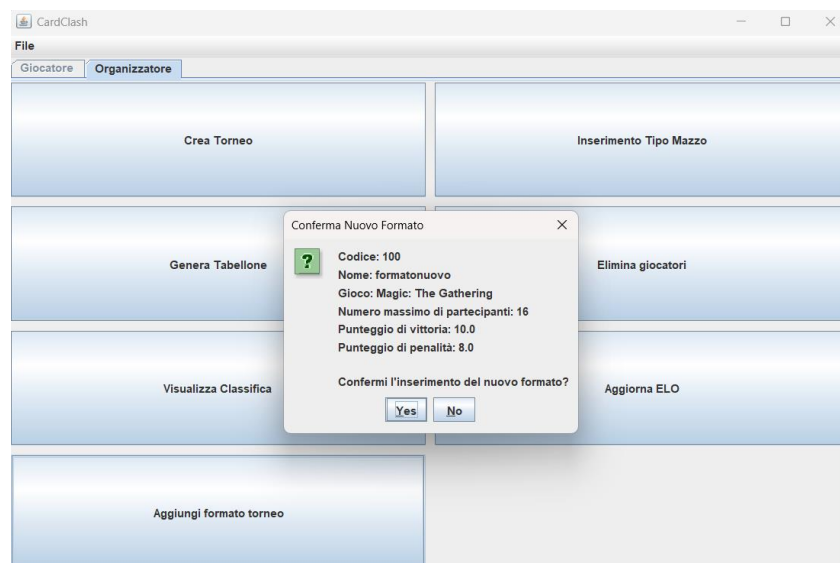


OK

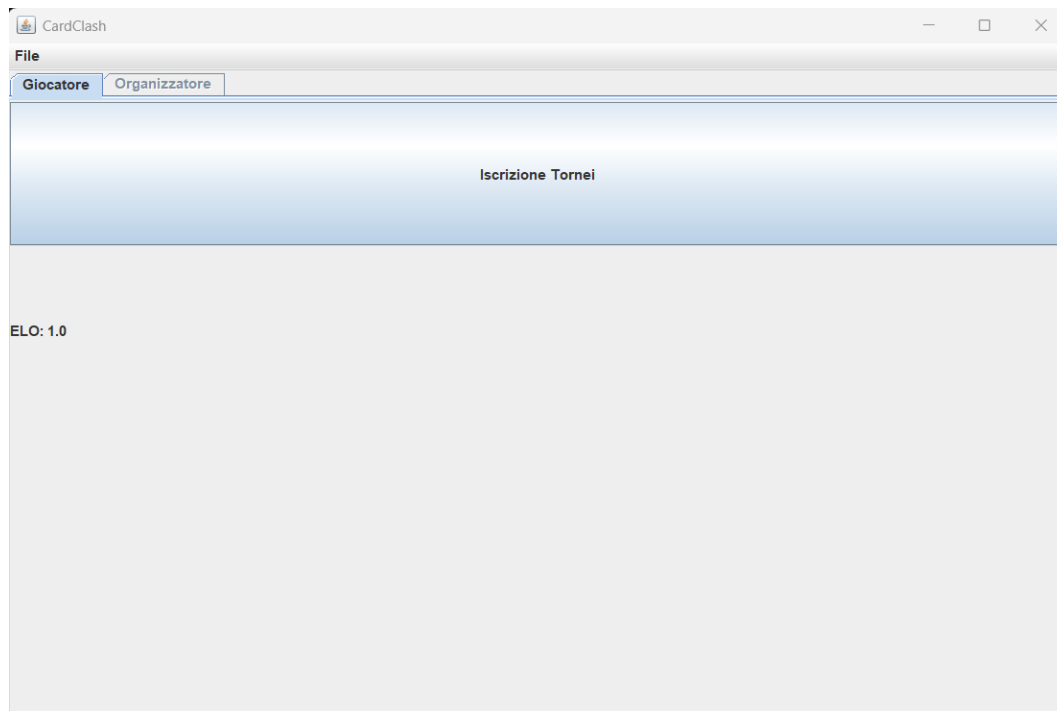
5.9 Popup del vincitore, dopo l'ultima eliminazione



5.10 Popup della conferma di creazione di un formato torneo



5.11 Interfaccia giocatore aggiornata dopo la partecipazione ad un torneo



6.0 Persistenza dei dati

In questa iterazione è stata implementata una classe apposita per la gestione della persistenza dei dati: `PersistenceHandler`. Si è voluta delegare la responsabilità della gestione del DOM di un file di persistenza xml, a questa classe, in accordo con il pattern Grasp Pure Fabrication. Dunque, essa non rappresenta un oggetto del modello di Dominio, ma è una classe utilizzata unicamente dalla GUI per caricare e salvare dati. Chiaramente si tratta di una persistenza embrionale e che non si appoggia ad un DBMS, ciò nonostante, essa è facilmente integrabile in versioni future dell'applicazione, tramite un refactoring unicamente rivolto a questa classe, senza intaccare altri componenti dell'applicazione, motivo per cui si è scelto di utilizzare questo pattern.

7.0 Test di sistema

Durante l'iterazione 4, una volta completata l'integrazione della GUI con il sistema esistente, si è proceduto all'esecuzione di test di sistema, lanciando l'applicazione e portando a termine i vari casi d'uso progettati. L'obiettivo era verificare la corretta integrazione della GUI e la sua interazione con le altre componenti del sistema, nonché la gestione degli errori e dei casi limite. A seguito di tali test non sono emersi errori critici e le prestazioni del sistema sono ottime, nonostante siano state riscontrate alcune imperfezioni e anomalie nell'esecuzione di particolari funzionalità.

Riportiamo qui di seguito, nel dettaglio, alcune delle funzionalità integrate tramite GUI e testate con un approccio black-box basato sui Casi d'Uso, mirato a seguire ogni fase dello scenario principale e delle sue estensioni al momento supportate e gestite:

- **Pannello di Creazione del Torneo (TournamentCreationPanel):**
 - **Verifica dell'obbligatorietà dei campi:** l'applicazione segnala un errore se uno o più campi (nome, data, orario, luogo) non vengono compilati.
 - **Validazione dei formati:** vengono mostrati messaggi di errore se la data non rispetta il formato "yyyy-MM-dd" o se l'orario non segue il formato "HH:mm".
 - **Controllo della validità della data:** viene impedita la creazione di un torneo se la data inserita è nel passato.
 - **Gestione dei conflitti:** se esiste già un torneo programmato per la stessa data, viene visualizzato un messaggio di errore.
 - **Selezione del formato:** il sistema presenta un menù a tendina con i formati disponibili; in caso di annullamento o selezione errata, il sistema notifica l'errore.

- **Processo di conferma:** una volta selezionato il formato, il pulsante “Conferma Creazione” diventa visibile e, al clic, il torneo viene creato, i campi vengono ripuliti e viene mostrata una conferma di successo.
- **Pannello dell’Organizzatore (OrganizerPanel):**
 - **Inserimento Tipo Mazzo:** l’organizzatore può selezionare un formato tramite un menù a tendina e inserire il nome della nuova tipologia di mazzo; se i dati non sono validi o la tipologia è già esistente, viene mostrato un messaggio di errore.
 - **Generazione del Tabellone:** è possibile selezionare un torneo futuro e generare il tabellone, che viene visualizzato in una tabella non modificabile. Se il numero di giocatori non è corretto o se l’operazione viene annullata, vengono visualizzati i messaggi di errore appropriati.
 - **Eliminazione Giocatori:** il sistema consente all’organizzatore di eliminare giocatori dal tabellone. Se l’e-mail inserita non corrisponde a nessun giocatore o se la rimozione non rispetta le regole (ad esempio, il numero dei giocatori rimasti non è una potenza di due), viene notificato l’errore. Inoltre, quando l’eliminazione porta alla conclusione del torneo, viene visualizzato il vincitore e il punteggio aggiornato.
 - **Visualizzazione della Classifica:** il sistema permette di selezionare un torneo e di visualizzare la classifica in una tabella non modificabile, con le colonne “Posizione”, “Email”, “Nickname” e “Punteggio”.
 - **Aggiornamento ELO:** sono stati implementati controlli (estensioni 8a e 8b) per impedire l’aggiornamento dell’ELO se il torneo non è ancora iniziato o se non è concluso. Solo quando il torneo è terminato, il sistema esegue l’aggiornamento e mostra una conferma.
 - **Aggiunta di un Nuovo Formato di Torneo:** l’organizzatore può creare un nuovo formato inserendo i dati richiesti (codice univoco, nome, gioco, numero massimo di partecipanti, punteggio di vittoria e di penalità). Dopo aver visualizzato le informazioni del formato creato, l’utente può confermare l’inserimento; in caso di conferma, il formato viene aggiunto con successo, altrimenti l’operazione viene annullata.

- **Pannello del Giocatore (PlayerPanel):**

- **Iscrizione a un Torneo:** il sistema presenta un menù a tendina con la lista dei tornei disponibili a cui il giocatore non è ancora iscritto. Se non ci sono tornei disponibili, viene mostrato un messaggio informativo.
- **Selezione del Torneo:** una volta selezionato il torneo, il sistema mostra un messaggio di conferma con il nome e la data del torneo scelto.
- **Inserimento del Nome del Mazzo:** il sistema richiede al giocatore di inserire il nome del mazzo; se il campo è vuoto, viene mostrato un errore.
- **Selezione della Tipologia di Mazzo:** il sistema mostra un menù a tendina con le tipologie di mazzo disponibili per il torneo selezionato. Se non ci sono tipologie disponibili, viene mostrato un messaggio di errore.
- **Conferma dell'Iscrizione:** dopo la selezione della tipologia di mazzo, il sistema chiede conferma prima di completare l'iscrizione. Se l'utente conferma, il sistema registra l'iscrizione e mostra un messaggio di successo; in caso contrario, l'operazione viene annullata.
- **Aggiornamento ELO in Tempo Reale:** il sistema aggiorna dinamicamente l'etichetta che mostra il punteggio ELO del giocatore ogni volta che viene modificato.

I suddetti testing black-box hanno consentito di verificare la robustezza e la correttezza dell'interfaccia utente, garantendo che il sistema risponda adeguatamente a una varietà di input e che gestisca errori e casi limite in modo chiaro e coerente. Questa fase di test ha reso possibile la scoperta e la correzione di alcune imperfezioni, migliorando la qualità complessiva dell'applicazione.

7.1 Revisione testing unitari

Infine, riportiamo una modifica effettuata alle classi di test *CardClashTest* e *TorneoTest*. Infatti sono stati aggiornati in maniera più precisa e completa, rispettivamente i seguenti metodi: *testMostraTorneiDisponibili* e *testIsAperto*. Tali modifiche non solo servono per coprire tutti gli scenari possibili di chiamata dei metodi, ma si pongono in linea con l'ultima versione del metodo *isAperto()* di *Torneo*, che, come detto in precedenza, è stato aggiornato in questa iterazione.