

Requisiti Funzionali e Non Funzionali:

Registrazione e Login

Requisiti Funzionali

- **RF1.1:** Il sistema deve permettere agli utenti di registrarsi creando un nuovo account personale.
- **RF1.2:** Durante la registrazione, l'utente deve poter inserire:
 - nome utente
 - nome
 - cognome
 - email
 - password
 - codice fiscale
- **RF1.3:** Il sistema deve validare i dati inseriti durante la registrazione:
 - nome utente: minimo 3 caratteri
 - password: minimo 8 caratteri, almeno una maiuscola e un numero
 - codice fiscale: formato corretto e corrispondenza con i dati anagrafici
 - età: almeno 18 anni

- **RF1.4:** Se i dati inseriti sono validi, il sistema deve:
 - generare un hash della password
 - salvare i dati nel database
 - inviare un messaggio di conferma della registrazione
- **RF1.5:** Il sistema deve permettere agli utenti registrati di accedere tramite nome utente/email e password.
- **RF1.6:** Se le credenziali sono corrette, l'utente viene autenticato e reindirizzato alla home.
- **RF1.7:** Se le credenziali sono errate, viene mostrato un messaggio di errore.
- **RF1.8:** L'utente autenticato deve poter accedere alle impostazioni dell'account.
- **RF1.9:** L'utente deve poter modificare la propria password inserendo:
 - la password attuale
 - la nuova password
- **RF2.0:** Il sistema deve verificare che:
 - la password attuale sia corretta
 - la nuova password rispetti i criteri di sicurezza
- **RF2.1:** Se la verifica ha esito positivo, il sistema deve aggiornare la password nel database e confermare l'aggiornamento.

- **RF2.2:** L'utente autenticato deve poter richiedere la cancellazione del proprio account.
- **RF2.3:** Il sistema deve:
 - verificare l'identità dell'utente
 - eliminare definitivamente i dati dal database
 - inviare conferma della cancellazione
- **RF2.4:** Se l'identità non è confermata o si verifica un errore tecnico, il sistema deve bloccare l'operazione e mostrare un messaggio di errore.

Requisiti Non Funzionali

- **RNF1:** Le password devono essere salvate in forma hashata con algoritmo sicuro.
- **RNF2:** Tutte le interazioni con il database devono essere sicure, affidabili e protette.
- **RNF3:** Tempi di risposta accettabili per tutte le operazioni.
- **RNF4:** Interfaccia chiara, semplice da usare e responsiva.
- **RNF5:** Controlli lato client e server per garantire integrità e validità dei dati.
- **RNF6:** Conformità alle normative sulla protezione dei dati personali.
- **RNF7:** Continuità del servizio salvo manutenzioni programmate.
- **RNF8:** Messaggi di errore e conferma comprensibili e informativi.

Gestione Giochi

Requisiti Funzionali

- **RF1.1:** Il Games_Admin può aggiungere nuovi giochi inserendo almeno il nome del gioco.
- **RF1.2:** Il Games_Admin può rendere un gioco visibile o non visibile ai giocatori.

Requisiti Non Funzionali

- **RNF1.1:** Operazioni in tempo reale senza ritardi percepibili.
- **RNF1.2:** Lista separata per giochi attivi e rimossi.
- **RNF1.3:** Interfaccia intuitiva e facile da usare.
- **RNF1.4:** Solo utenti autenticati come Games_Admin possono accedere alla gestione.
- **RNF1.5:** Gestione efficiente di un numero crescente di giochi.

Gestione Amministratori

Requisiti Funzionali

- **RF1.1:** Login SuperAdmin con email e password predefiniti.
- **RF1.2:** Email admin valida (formato: nome.cognome01@domini.com).

- **RF1.3:** Login Admin Specifici con email e password salvati nel database.
- **RF1.4:** In caso di credenziali errate, mostrare messaggio di errore.
- **RF1.5:** Dashboard dedicata per ogni Admin dopo il login.
- **RF1.6:** Il SuperAdmin può creare un nuovo admin inserendo:
 - nome
 - cognome
 - tipologia di Admin
- **RF1.7:** Il SuperAdmin può visualizzare la lista degli admin creati.
- **RF1.8:** L'admin può effettuare login con le credenziali assegnate.
- **RF1.9:** Interfaccia personalizzata per admin autenticati.
- **RF2.0:** Gli admin possono visualizzare i propri dati.
- **RF2.1:** Gli admin possono modificare la propria password.
- **RF2.2:** Gli admin possono effettuare logout.

Requisiti Non Funzionali

- **RNF1.1:** Uso del pattern MVC.
- **RNF1.2:** Uso del Factory Pattern per creazione admin.
- **RNF1.3:** Uso di Swing per l'interfaccia.
- **RNF1.4:** Uso di CardLayout per la gestione schermate.

- **RNF1.5:** Azioni eseguibili con pulsanti intuitivi.
- **RNF1.6:** Messaggi chiari e visibili.
- **RNF1.7:** Validazione credenziali prima dell'invio.
- **RNF1.8:** JPasswordField per input password.
- **RNF1.9:** Programma leggero e reattivo.
- **RNF2.0:** Password generata automaticamente tramite PasswordGenerator.

Gestione del Wallet e delle Transazioni

Requisiti Funzionali

- **RF3.1:** Accesso alla sezione Wallet per utenti autenticati.
- **RF3.2:** Deposito fondi (importo + metodo pagamento).
- **RF3.3:** Prelievo fondi (importo + dettagli conto).
- **RF3.4:** Metodi di pagamento supportati:
 - Carta
 - PayPal
 - Apple Pay
 - Satispay
- **RF3.5:** Inoltro richieste al gateway e gestione esito.

- **RF3.6:** Aggiornamento saldo e conferma dopo transazione positiva.
- **RF3.7:** Messaggio di errore per transazioni fallite.
- **RF3.8:** Visualizzazione storico transazioni.
- **RF3.9:** Gestione limiti di deposito:
 - Giornaliero: €1000
 - Mensile: €10.000
- **RF3.10:** Visualizzazione storico giocate e depositi.
- **RF3.11:** Tabelle separate per transazioni e giocate.
- **RF3.12:** Wallet_Admin può gestire metodi di pagamento.
- **RF3.13:** Attivazione/disattivazione metodi pagamento tramite flag.
- **RF3.14:** Modifiche visibili in tempo reale agli utenti.

Requisiti Non Funzionali

- **RNF1:** Comunicazioni sicure con gateway (HTTPS + crittografia).
- **RNF2:** Tempo massimo risposta: 3 secondi.
- **RNF3:** Transazioni salvate con timestamp.
- **RNF4:** Interfaccia wallet user-friendly e responsive.
- **RNF5:** Conformità alle normative vigenti.
- **RNF6:** Modifiche del Wallet_Admin riflesse subito.

- **RNF7:** Notifiche chiare per ogni operazione.
- **RNF8:** Limiti non modificabili oltre le soglie massime.
- **RNF9:** Protezione dati tramite autenticazione sicura.

Gestione delle Competizioni

Requisiti Funzionali

- **RF1.1:** Creazione torneo con:
 - nome torneo
 - quota iscrizione
 - livello
 - max giocatori
 - tipo giochi
 - numero round
 - fish per round
 - tempo per round
- **RF1.2:** Torneo visibile nella sezione "Tornei".
- **RF1.3:** Admin può modificare o cancellare un torneo.
- **RF1.4:** Iscrizione utente solo se:

- login effettuato
- saldo sufficiente
- punteggio minimo (se richiesto)
- **RF1.5:** Conferma iscrizione visibile nella propria area.
- **RF1.6:** Annullamento iscrizione prima dell'inizio.
- **RF1.7:** Abbandono torneo in corso (segnato come "Cancelled").

Requisiti Non Funzionali

- **RNF1.1:** Classifiche aggiornate in tempo reale.
- **RNF1.2:** Risposte rapide per operazioni torneo.
- **RNF1.3:** Coerenza tra azioni utente e dati visualizzati.
- **RNF1.4:** Informazioni torneo chiare (data, orari, tempo rimanente).
- **RNF1.5:** Iscrizione doppia o non valida bloccata.
- **RNF1.6:** Stato torneo ben visibile (Non iniziato, In corso, Concluso).
- **RNF1.7:** Sicurezza e tracciabilità delle azioni utente.

Servizio di Helpdesk

Requisiti Funzionali

- **RF1.1:** Creazione ticket con titolo e conferma.

- **RF1.2:** Accesso alla chat del ticket tramite menu a tendina.
- **RF1.3:** Invio/lettura messaggi nella chat.
- **RF1.4:** Admin può prendere in carico un ticket.
- **RF1.5:** Accesso alla chat dei ticket presi in carico.
- **RF1.6:** Admin può chiudere un ticket (blocco chat).
- **RF1.7:** Admin può visualizzare transazioni, giocare e rimborsare.
- **RF1.8:** Messaggi e rimborsi aggiornati in tempo reale.

Requisiti Non Funzionali

- **RNF1.1:** Aggiornamento in tempo reale senza ricaricare la pagina.
- **RNF1.2:** Tempi di risposta brevi per tutte le operazioni.
- **RNF1.3:** Consistenza tra azioni e informazioni visualizzate.
- **RNF1.4:** Data e ora dei messaggi ben visibili e formattate.
- **RNF1.5:** Chat bloccata per ticket chiusi, con messaggio informativo.
- **RNF1.6:** Stato ticket sempre ben visibile (Aperto, In corso, Chiuso).