

Progetto APL – Ideazione

SOCCKER - BET

Si vuole realizzare un sistema di betting on line, che permetta agli utenti sottoscritti al sistema di poter piazzare scommesse calcistiche attraverso il portale. Il sistema sarà composto da tre entità: Client che si occuperà dell'interazione con le utenze e la visualizzazione dell'interfaccia grafica, il Server che si occuperà delle operazioni di autenticazione degli utenti, validazione delle scommesse e interazione con il database (MongoDB) per la gestione della persistenza e lo Stats Server che si occuperà di tener traccia di statistiche, interagendo con il DB, che potranno essere visualizzate dall'utente.

Sono previste due tipologie di utenze:

- Admin: deve potersi loggare, si occupa della gestione (inserimento e rimozione) degli eventi e quote, deve poter visualizzare informazioni relative al sistema (stato di funzionamento del sistema e informazioni statistiche relative alle vittorie e giocate);
- Utente: deve potersi registrare/loggare al sistema attraverso la GUI, visualizzare le proprie informazioni personali e relative al proprio profilo (dati personali, storico giocate e saldo), e poter piazzare la scommessa.

Il client deve:

- Visualizzare la GUI;
- Fornire la pagina di login/logout, raccogliere le informazioni inserite dall'utente e inviarle tramite REST al server;
- Fornire la pagina di registrazione, come sopra;
- Permettere all'utente di selezionare gli eventi su cui vuole scommettere;
- Prenotare la giocata scegliendo l'importo da scommettere, inviare una richiesta di validazione della giocata al server, deve poter eseguire un controllo sull'importo minimo da giocare (ed eventualmente se l'importo giocato è maggiore del saldo);
- Visualizzare le informazioni personali dell'utente e di modificare i dati personali.

Il server deve:

- Gestire il DB e l'interazione con esso e rendere disponibili i dati memorizzati agli utenti loggati (privilegi differenti a seconda del ruolo);
- Eseguire le operazioni di autenticazione, verificare che le credenziali fornite nella fase di login siano corrette e rispondere al client inviando gli eventi disponibili;
- Eseguire le operazioni di memorizzazioni degli utenti, e degli eventi immessi nel sistema;
- Eseguire le operazioni di validazione delle giocate, verificando la validità degli eventi selezionati (data e ora conforme), e rispondere al client inviando una ricevuta dell'avvenuta validazione;
- Verificare la validità degli eventi disponibili nel DB ed aggiornare i nuovi eventi resi disponibili al client (aggiornamento della view in maniera asincrona).

Client implementato in python, con libreria QT. Server implementato in C++, multithread, ogni richiesta del client verrà gestita eseguendo uno specifico thread. Comunicazione bidirezionale attraverso REST.

Soluzioni tecnologiche:

client: PyQt, python

DB: MongoDB, mongocxx (per l'integrazione)

server: C++, RESTinio (creare server rest in c++)

Utilizzo di un token per la gestione della sessione, ed avere accesso alle funzionalità del sistema.

Documenti necessaria da creare e memorizzare nel DB:

- **Utenti:** documento degli utenti registrati al sito, contiene le informazioni relative all'utente e al suo profilo, presenta i seguenti attributi.
 - Id_utente: identificativo univoco autoincrement (Primary Key),
 - Nome: stringa not null
 - Cognome: stringa not null
 - Email: stringa not null unique
 - Password: stringa not null,
 - Saldo: float default 0.00,
 - Admin: bool default false.

- **Eventi:** documento che raccoglie gli eventi e i relativi dati e quote su cui poter scommettere, presenta i seguenti attributi:
 - id_evento: identificativo univoco autoincrement (Primary Key);
 - nome: stringa not null;
 - data_inizio: Date not null;
 - data_fine: Date not null;
 - 1: float default 1.00;
 - 2: float default 1.00;
 - X: float default 1.00;
 - GG: float default 1.00;
 - NG: float default 1.00;
 - Over_2.5: float default 1.00;
 - Under_2.5: float default 1.00.
- **Scommesse:**
 - Id_scommessa: int univoco autoincrement (Primary Key);
 - Id_utente: reference a id_utente di Utenti;
 - Esiti eventi [esito eventi: riferimento a Esito_Evento];
 - Importo scommesso: float;
 - Potenziale vincita: float.
- **Scommesse Vinte:**
 - Id_scommessa: riferimento alla scommessa;
 - Id_utente: reference a id_utente di Utenti;
 - Vincita: float;
 - Vinta: Bool.
- **Statistiche_sistema:**
 - Numero utenti registrati: int;
 - Numero scommesse nel sistema: int;
 - Numero scommesse vinte: int;
 - Numero scommesse perse: int;
 - Entrate: float;
 - Uscite: float.
- **Statistiche_utente:**
 - Id_utente: reference a id utente di Utenti;
 - Denaro ricaricato: int;
 - Denaro scommesso: int;
 - Denaro vinto: float;

- Numero scommesse vinte: int;
- Numero scommesse perse: int.

Attore	Obiettivo	UC
Utente/Admin	Gestione del login/logout/registrazione degli utenti.	UC1: Login/Registrazione
Utente	Visualizzazione e modifica dei dati relativi all'utente loggato.	UC2: Gestione info profilo
Utente	Prenotazione della scommessa a seguito della scelta degli esiti per i vari eventi.	UC3: Prenotazione scommessa
Admin	Aggiunta e rimozione degli eventi; modifica delle date e delle quote relative ad essi.	UC4: Modifica eventi e quote
Admin	-verifica delle scommesse vinte/perse -visualizzazione delle statistiche di sistema	UC5: Gestione sistema